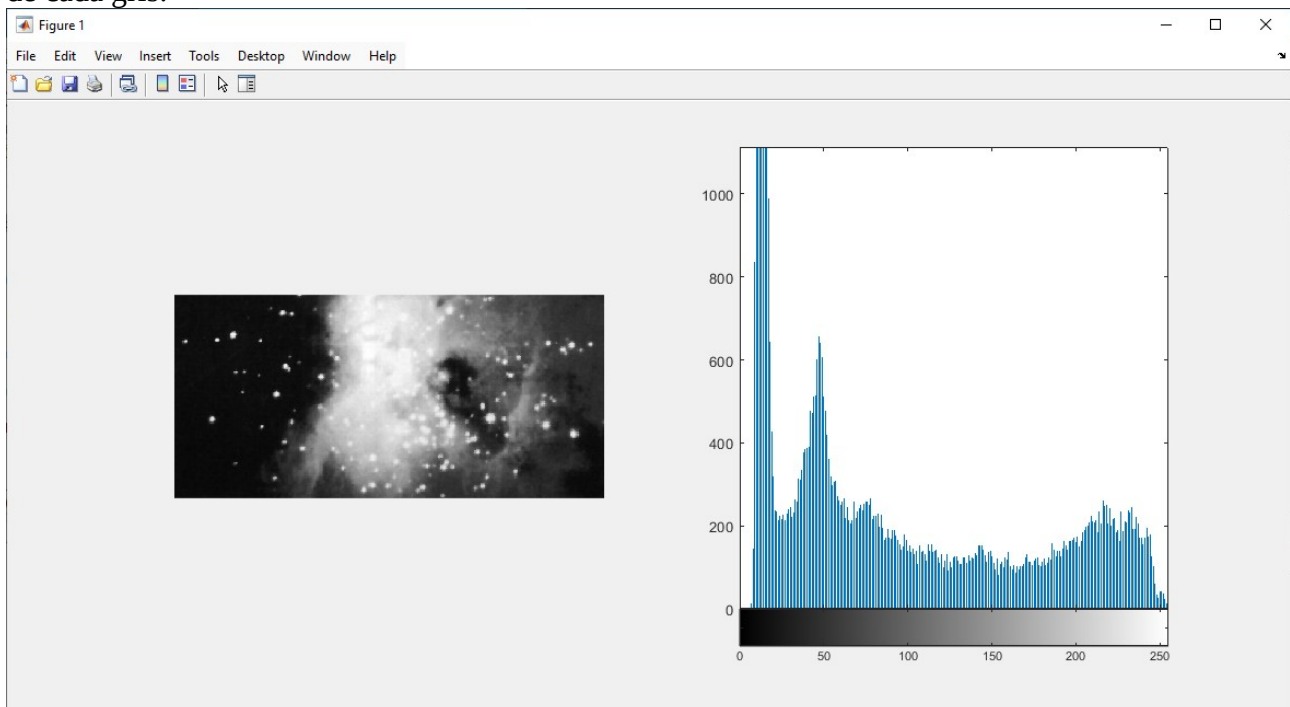


Practica 3

Apartado 4.

```
%Carga la imagen  
I=imread(' ../Imagenes/ngc4024l.tif');  
%Muestra la imagen  
imshow(I)  
%Muestra el histograma de la imagen  
figure, imhist(I)
```

Usando el comando “imhist” podemos crear el histograma de cualquier imagen y ver la frecuencia de cada gris:



En la imagen de la nebulosa vemos que tiende mas a los colores oscuros, cosa que vemos reflejado en el histograma.

Apartado 5.

A.

```
%Carga la imagen y la muestro
I=imread(' ../Imagenes/ngc40241.tif');
subplot(3,1,1), imshow(I);
%Transformo la imagen a valores entre [0,1] y creo la segunda imagen
I1=im2double(I);
I2=I1;
%Creo los bucles para recorrer la imagen pixel a pixel
for n=1:size(I,2),
    for m=1:size(I,1),
        %Multiplico el valor de cada pixel por 255/40
        %Por lo que cualquier valor por encima de 40 sera igual a 1
        I2(m,n)=I1(m,n)*255/40;
        %Si es mayor que 1, lo igualamos a 1
        if I2(m,n)>1,
            I2(m,n)=1;
        end
    end
end
subplot(3,1,2), imshow(I2);

%Con imadjust podemos tambien hacerlo pasandole primero la imagen, como
%segundo argumento el rango de tonos de gris que queremos ajustar y tercero
%el rango al que queremos ajustarlo.
J=imadjust(I,[0 (40/255)],[]);
subplot(3,1,3), imshow(J);
```

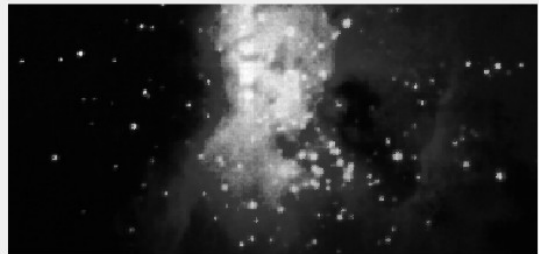
Como se puede ver en el resultado, aumenta el brillo en la imagen y aumentan las zonas claras.



B.

```
%Carga la imagen y la muestro
I=imread('../Imagenes/ngc40241.tif');
subplot(1,2,1), imshow(I);
%Transformo la imagen a rango [0,1]
I1=im2double(I);
%Factor de reduccion del contraste
a=0.3 ;
%Punto donde cambiamos la transformacion
c=190/255 ;
for n=1:size(I,2)
    for m=1:size(I,1)
        %Si el pixel tiene una intensidad menor a c
        %Lo multiplicamos por a reduciendo el contraste
        if I1(m,n)<c
            I2(m,n)=I1(m,n)*a;
        %En otro caso, aplicamos esa formula aumentando el contraste
        else
            I2(m,n)=a*c+(I1(m,n)-c)*(1-a*c)/(1-c);
        end
    end
end
subplot(1,2,2), imshow(I2);
```

En este caso aumentan las zonas oscuras aumentando el contraste entre las zonas mas claras y oscuras.



C.

```
% Cargar la imagen
I = imread(' ../Imagenes/ngc40241.tif');
I = im2double(I); % Convertir a rango [0,1]
subplot(1,2,1), imshow(I);

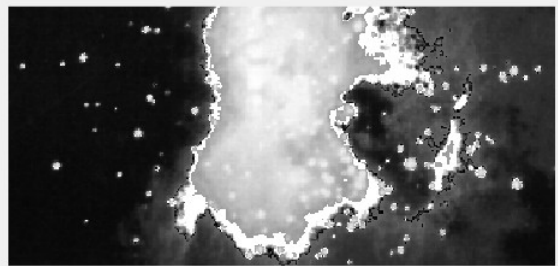
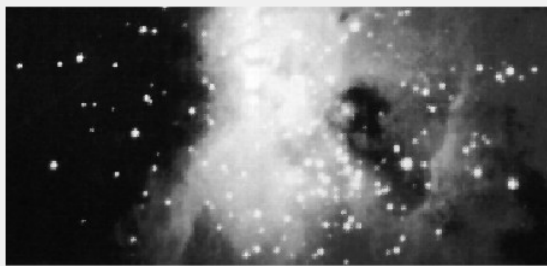
c1 = 0.4; % Límite inferior
c2 = 0.6; % Límite superior
a = 2; % Factor de aumento de contraste

% Inicializar la imagen transformada
I2 = I;

for m = 1:size(I,1)
    for n = 1:size(I,2)
        %Si el valor del pixel esta entre [c1, c2] estiramos el valor
        %Si es igual a c1 el valor sera 0 y si (c1+c2)/2 sera 1
        if I(m,n) >= c1 && I(m,n) <= c2
            I2(m,n) = a * (I(m,n) - c1) / (c2 - c1);
        end
    end
end

subplot(1,2,2), imshow(I2);
```

En este caso solo afectamos a los pixeles dentro del intervalo [0.4, 0.6]. Que aplicando la formula tenemos que sean 0 o 1 prácticamente por lo que conseguimos el contorno de las zonas mas brillantes.



D.

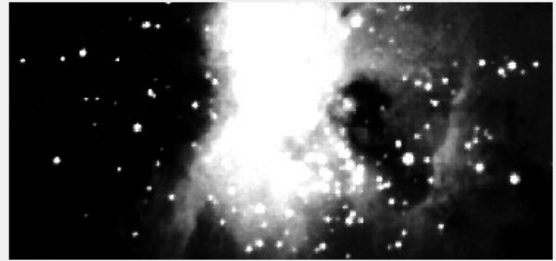
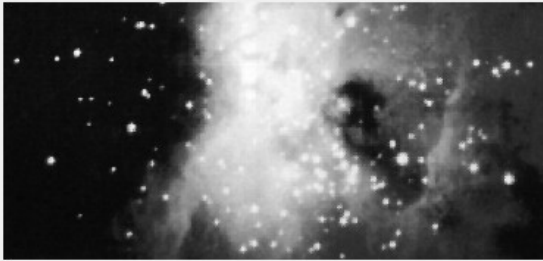
```
% Cargar la imagen
I = imread('../Imagenes/ngc40241.tif');
I = im2double(I); % Convertir a rango [0,1]
subplot(1,2,1), imshow(I);

% Factor de realce
a = 1.5;

% Aplicamos  $T(x) = ax^2$ 
% Usamos .^2 para elevar cada valor de la matriz a 2
I2 = a * I.^2;

subplot(1,2,2), imshow(I2);
```

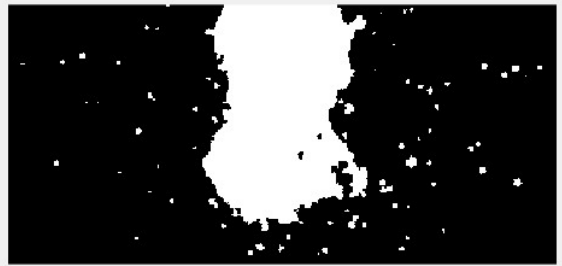
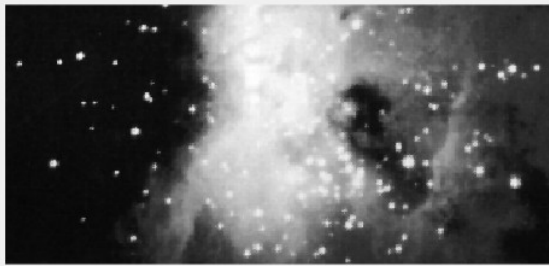
En este caso realzamos las intensidades de los pixeles por lo que los pixeles mas oscuros serán aun mas oscuros y los mas claros aun mas claros.



E.

```
% Cargar la imagen y la muestra  
I = imread('../Imagenes/ngc40241.tif');  
subplot(1,2,1), imshow(I);  
% Convertir a rango [0,1]  
I = im2double(I);  
  
% Definir el umbral para detectar los objetos brillantes  
% Este valor dependera de la imagen  
threshold = 0.7;  
  
% Creamos la imagen binario, como este caso los colores van a ser 0 o 1.  
% Podemos simplemente comparar la matriz contra el valor de threshold y nos  
% quedamos con los valores booleanos que al final seran 0 o 1  
I_bin = I > threshold;  
  
subplot(1,2,2), imshow(I_bin);
```

En este caso binarizamos la imagen por lo que pasamos de una imagen de grises a solo blanco y negro quedándonos solo con las zonas mas brillantes.



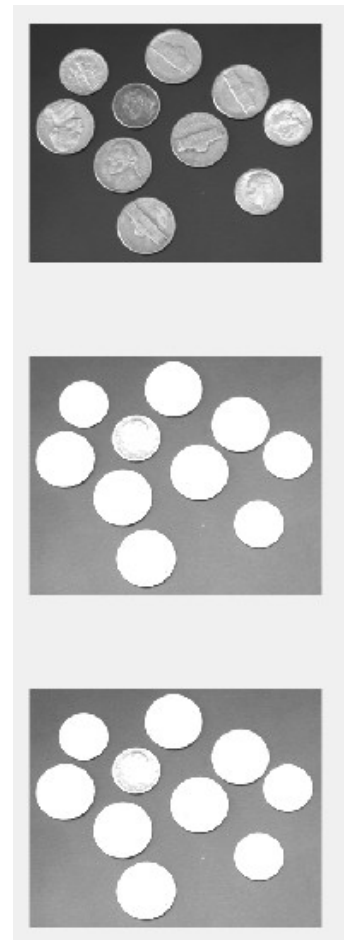
Apartado 5 con otra imagen

A.

```
%Cargo la imagen y la muestro
I=imread(' ../Imagenes/coins.png');
subplot(3,1,1), imshow(I);
%Transformo la imagen a valores entre [0,1] y creo la segunda imagen
I1=im2double(I);
I2=I1;
%Creo los bucles para recorrer la imagen pixel a pixel
for n=1:size(I,2),
    for m=1:size(I,1),
        %Multiplico el valor de cada pixel por 255/120
        %Por lo que cualquier valor por encima de 120 sera igual a 1
        I2(m,n)=I1(m,n)*255/120;
        %Si es mayor que 1, lo igualamos a 1
        if I2(m,n)>1,
            I2(m,n)=1;
        end
    end
end
subplot(3,1,2), imshow(I2);

%Con imadjust podemos tambien hacerlo pasandole primero la imagen, como
%segundo argumento el rango de tonos de gris que queremos ajustar y tercero
%el rango al que queremos ajustarlo.
J=imadjust(I,[0 (120/255)],[]);
subplot(3,1,3),imshow(J);
```

Para este caso he elegido el valor 120 de manera que al aplicar las transformaciones se pierdan los detalles de las monedas.



B.

```
%Carga la imagen y la muestra
I=imread(' ../Imagenes/coins.png');
subplot(1,2,1), imshow(I);
%Transformo la imagen a rango [0,1]
I1=im2double(I);
%Factor de reduccion del contraste
a=0.1 ;
%Punto donde cambiamos la transformacion
c=80/255 ;
for n=1:size(I,2)
    for m=1:size(I,1)
        %Si el pixel tiene una intensidad menor a c
        %Lo multiplicamos por a reduciendo el contraste
        if I1(m,n)<c
            I2(m,n)=I1(m,n)*a;
        %En otro caso, aplicamos esa formula aumentando el contraste
        else
            I2(m,n)=a*c+(I1(m,n)-c)*(1-a*c)/(1-c);
        end
    end
end
subplot(1,2,2), imshow(I2);
```

En este caso, con los valores que he elegido borro el fondo y lo dejo del mismo tono de negro.



C.

```
% Cargar la imagen
I = imread(' ../Imagenes/coins.png');
I = im2double(I); % Convertir a rango [0,1]
subplot(1,2,1), imshow(I);

c1 = 0.0; % Límite inferior
c2 = 0.4; % Límite superior
a = 2; % Factor de aumento de contraste

% Inicializar la imagen transformada
I2 = I;

for m = 1:size(I,1)
    for n = 1:size(I,2)
        %Si el valor del pixel esta entre [c1, c2] estiramos el valor
        %Si es igual a c1 el valor sera 0 y si (c1+c2)/2 sera 1
        if I(m,n) >= c1 && I(m,n) <= c2
            I2(m,n) = a * (I(m,n) - c1) / (c2 - c1);
        end
    end
end

subplot(1,2,2), imshow(I2);
```

Aquí algo parecido pero dejando el fondo en blanco, aunque afecta un poco a las monedas seguimos teniendo los contornos.



D.

```
% Cargar la imagen
I = imread(' ../Imagenes/coins.png');
I = im2double(I); % Convertir a rango [0,1]
subplot(1,2,1), imshow(I);

% Factor de realce
a = 1.5;

% Aplicamos T(x) = ax^2
% Usamos .^2 para elevar cada valor de la matriz a 2
I2 = a * I.^2;

subplot(1,2,2), imshow(I2);
```

Usando el factor 1.5 puedo hacer que el fondo sea del mismo color, negro en este caso.



E.

```
% Cargar la imagen y la muestro
I = imread('./Imagenes/coins.png');
subplot(1,2,1), imshow(I);
% Convertir a rango [0,1]
I = im2double(I);

% Definir el umbral para detectar los objetos brillantes
% Este valor dependera de la imagen
threshold = 0.35;

% Creamos la imagen binario, como este caso los colores van a ser 0 o 1.
% Podemos simplemente comparar la matriz contra el valor de threshold y nos
% quedamos con los valores booleanos que al final seran 0 o 1
I_bin = I > threshold;

subplot(1,2,2), imshow(I_bin);
```

Aquí uso 0.35 para que el fondo y las monedas sean de distinto color.

