

Laboratorio calificado n°1

Tema:

Transformaciones básicas a nivel espacial

Curso:

Procesamiento Digital de Imágenes y visión artificial

Docente:

Marina Gabriela Sadith Perez Paredes

Alumnos:

Morales Máximo, Diana Katherine u18210750

Otoya Morales, Oscar Felix Antonio 1630618

Espinoza Valera, Jesús Alberto Francisco U17102566

Zuñiga Lujan, Edwin Alexander U17204235

Tizon Alegre, Christian Paolo 1635663

2021

## GUÍA N° 1 – TRANSFORMACIONES BÁSICAS A NIVEL ESPACIAL

FACULTAD	CURSO	AMBIENTE
Ingeniería	Procesamiento de imágenes y visión artificial	LABORATORIO REMOTO

ELABORADO POR	KEVIN ACUÑA CONDORI ALBERTO ALVARADO RIVERA	APROBADO POR	JAVIER PIÉROLA
VERSIÓN	001	FECHA DE APROBACIÓN	27/08/2020

### 1. LOGRO GENERAL DE LA UNIDAD DE APRENDIZAJE

- ☐ Conoce las diferentes aplicaciones actuales del procesamiento de imágenes, asimismo analiza los fundamentos y la formación de las imágenes digitales para su posterior procesamiento y transformación.

### 2. OBJETIVOS ESPECÍFICOS DE LA PRÁCTICA

- ☐ Al final de la sesión el estudiante conoce las principales transformaciones básicas a nivel espacial de imágenes digitales.

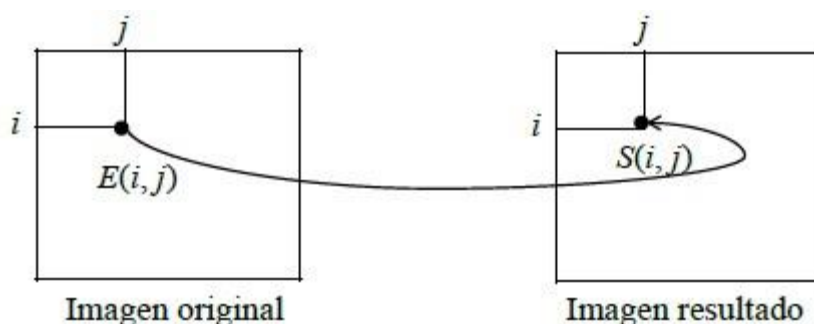
### 3. MATERIALES Y EQUIPOS

- ☐ Software Octave 5.2.0 o superior (free)

## 5. FUNDAMENTO

La transformación de una imagen consiste en modificar el contenido de la misma con un objetivo concreto, como puede ser el de prepararla para un posterior análisis. Se pueden distinguir dos tipos de transformaciones: basadas en los niveles de intensidad de la imagen, o como consecuencia de la aplicación de una operación geométrica. Dentro del primer grupo se consideran por un lado las transformaciones que son consecuencia de la aplicación de una función sobre el valor de intensidad de cada píxel individualmente, y las transformaciones en las que los píxeles vecinos intervienen en la misma con distintas finalidades. En la presente guía nos ocupamos de las transformaciones sobre el valor de intensidad de cada píxel individual a nivel espacial.

Estas operaciones tienen la particularidad de transformar la imagen mediante la modificación uno a uno de los píxeles de la imagen; es decir, el valor de intensidad del píxel  $S(i,j)$  de la imagen de salida es el resultado de aplicar una determinada transformación sobre el valor de intensidad del píxel  $E(i,j)$  de la imagen original, tal y como se puede apreciar en la siguiente figura



Basándonos en la imagen podemos comprender una transformación como una transformación matemática sobre la imagen original hacia la imagen resultante, es decir:  $S(i,j)=f(E(i,j))$

En la siguiente tabla se muestran los operadores de transformación básica a nivel espacial más empleados para el procesamiento digital de imágenes.

Operador	Operación
Operador identidad	$S(i,j) = E(i,j)$
Operador inversor	$S(i,j) = 255 - E(i,j)$
Operador umbral	$S(i,j) = \begin{cases} 0, & E(i,j) \leq p \\ 1, & E(i,j) > p \end{cases}$
Operador intervalo de umbral binario	$S(i,j) = \begin{cases} 0, & p_1 < E(i,j) < p_2 \\ 1, & E(i,j) \leq p_1 \text{ o } E(i,j) \geq p_2 \end{cases}$
Operador intervalo de umbral binario invertido	$S(i,j) = \begin{cases} 1, & p_1 < E(i,j) < p_2 \\ 0, & E(i,j) \leq p_1 \text{ o } E(i,j) \geq p_2 \end{cases}$
Operador umbral de la escala de grises.	$S(i,j) = \begin{cases} E(i,j), & p_1 < E(i,j) < p_2 \\ 1, & E(i,j) \leq p_1 \text{ o } E(i,j) \geq p_2 \end{cases}$

## 6. PROCEDIMIENTO (DESARROLLO DE LA PRÁCTICA)

En esta primera etapa se procederá a brindar un ejemplo basado en la figura 1 que podrá encontrar en los archivos adjuntos (subidos en CANVAS) con el nombre “lab1fig1.jpg”.



Figura 1. Fresas (imagen de prueba)

Los pasos a seguir para el ejemplo son los siguientes:

1. Ingresar a Octave y posicionar las ventanas de modo tal que tengamos la distribución recomendada en la figura 2. Recordar que para visualizar la ventana “Editor” se puede realizar ingresando el comando “edit” en la ventana de “command Windows”.

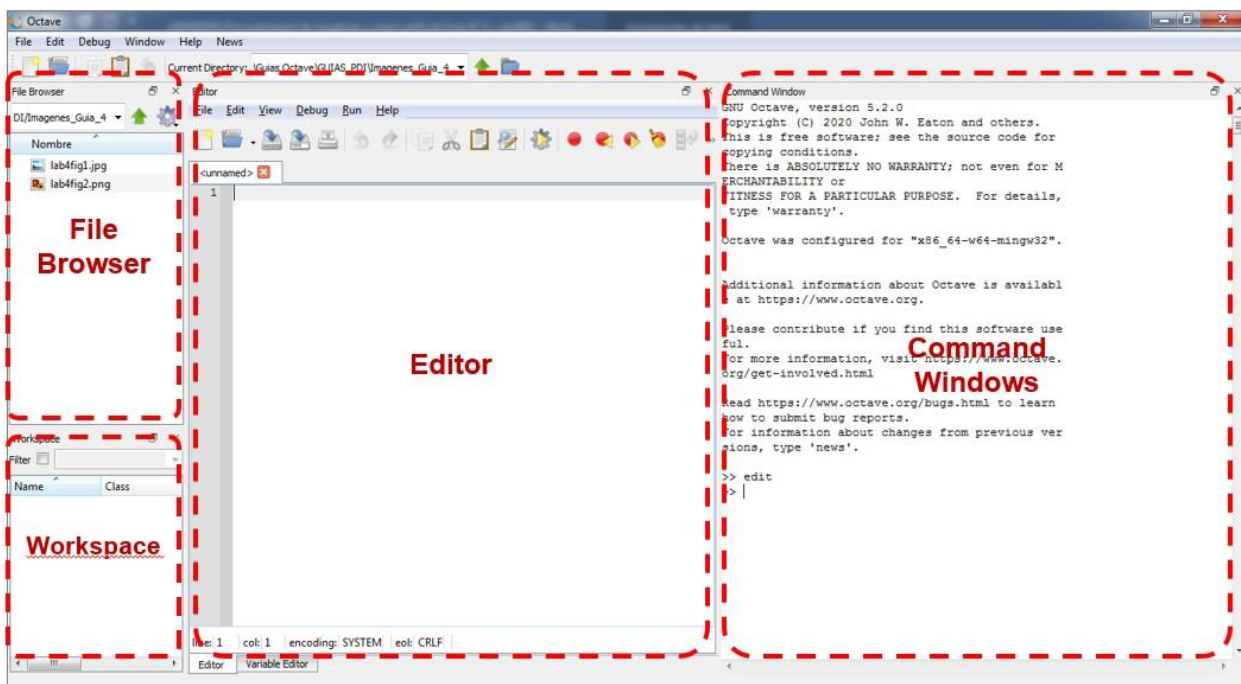
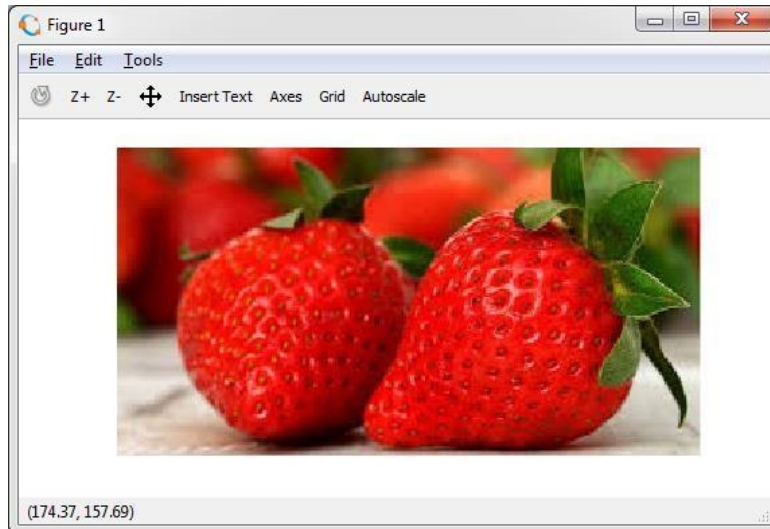


Figura 2. Ubicación de las ventanas

2. Leer la imagen del ejemplo, para ello es importante emplear el comando "imread", una vez leída la imagen proceder a visualizarla empleando el comando "imshow".

```
1 clc
2 clear all
3 close all
4
5 % Lectura de la imagen
6 I = imread("lab4fig1.jpg"); % almacenar la imagen en la variable I
7
8 % Visualizar la imagen
9 figure(1) % crear ventana
10 imshow(I) % mostrar la figura
```

Resultado:



3. Obtener una imagen a escala de grises basado en la imagen RGB leída

```
1 clc
2 clear all
3 close all
4
5 % Lectura de la imagen
6 I = imread("lab4fig1.jpg"); % almacenar la imagen en la variable I
7 Igris = rgb2gray(I); % pasar a escala de grises
8 Ir = I(:, :, 1); % Capa Roja
9 Ig = I(:, :, 2); % Capa Verde
10 Ib = I(:, :, 3); % Capa Azul
11
12 % Visualizar la imagen
13 figure(1) % crear ventana
14 subplot(2,3,1) % mostrar la figura rgb
15 imshow(I)
16 title("RGB")
17 subplot(2,3,2) % mostrar la figura gris
18 imshow(Igris)
19 title("Gris Ponderada")
20 subplot(2,3,4) % mostrar la capa roja
21 imshow(Ir)
22 title("Capa Roja")
23 subplot(2,3,5) % mostrar la capa verde
24 imshow(Ig)
25 title("Capa Verde")
26 subplot(2,3,6) % mostrar la capa azul
27 imshow(Ib)
```

Resultado:



4. Realizar una transformación umbral con parámetro  $p=$  a la capa roja.

```
30 % Transformación umbral
31 Iu = Ir < 70;
32 figure(2)
33 subplot(1,3,1)
34 imshow(I)
35 subplot(1,3,2)
36 imshow(Ir)
37 subplot(1,3,3)
38 imshow(Iu);
39
```

Resultado:





## Parte para desarrollar:

Trabajando con esta imagen RGB:



Figura 1 Imagen RGB

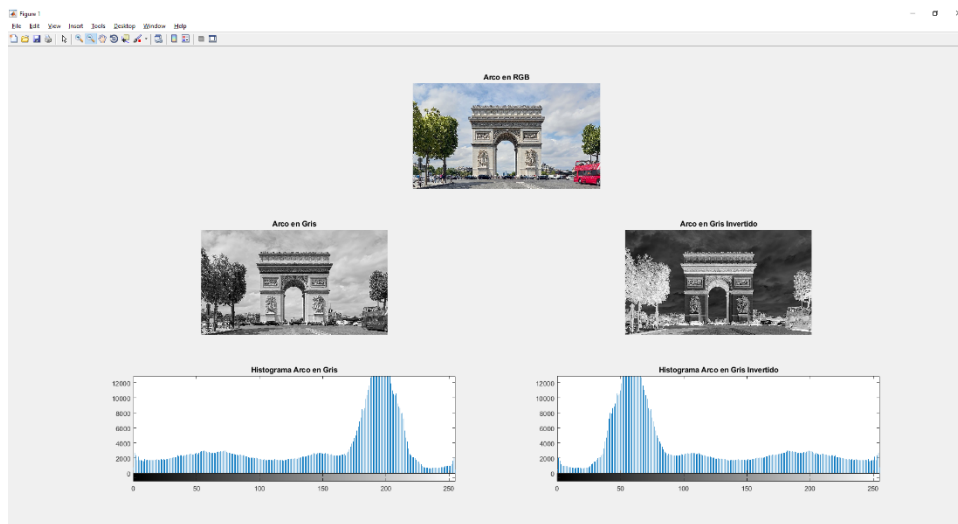
Este es el código aplicado a la imagen anterior para desarrollar las preguntas 1,2 y 3

```
clc, clear all, close all

imagen = imread('arco.jpg');
imagen_gray = rgb2gray(imagen);
imagen_gray_inv = imcomplement(imagen_gray);
% Pero tambien se puede realizar haciendo una resta: 255 - imagen_gray

$Parte 1/2/3
figure(1)
subplot(3,2,1:2), imshow(imagen),           title('Arco en RGB')
subplot(3,2,3), imshow(imagen_gray),        title('Arco en Gris')
subplot(3,2,4), imshow(imagen_gray_inv),    title('Arco en Gris Invertido')
subplot(3,2,5), imhist(imagen_gray),        title('Histograma Arco en Gris')
subplot(3,2,6), imhist(imagen_gray_inv),    title('Histograma Arco en Gris Invertido')
```

Y esto fue lo que conseguimos:



Arco en Gris



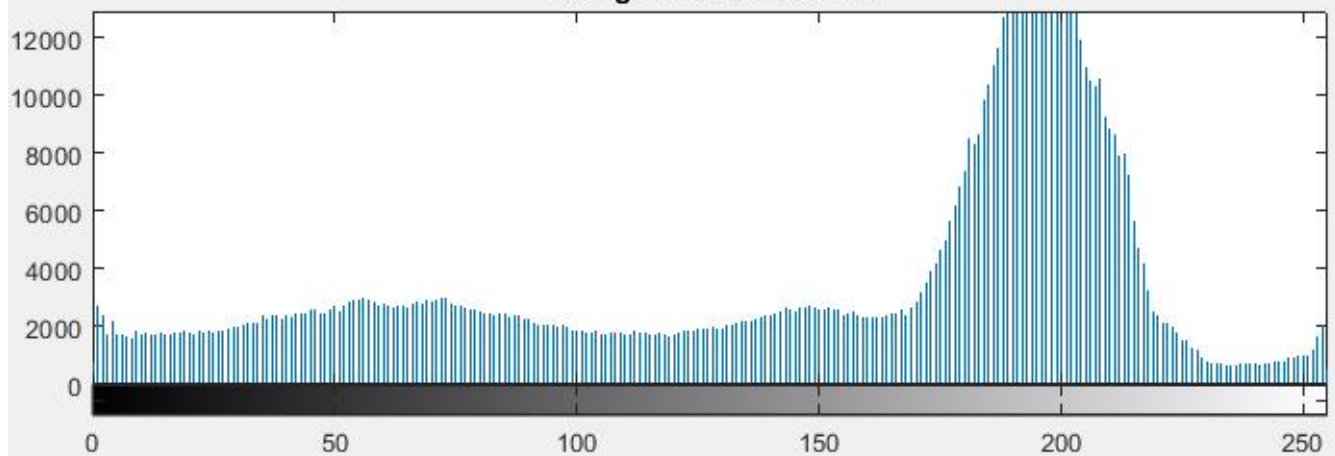
Arco en Gris Invertido



Arco en Gris



Histograma Arco en Gris

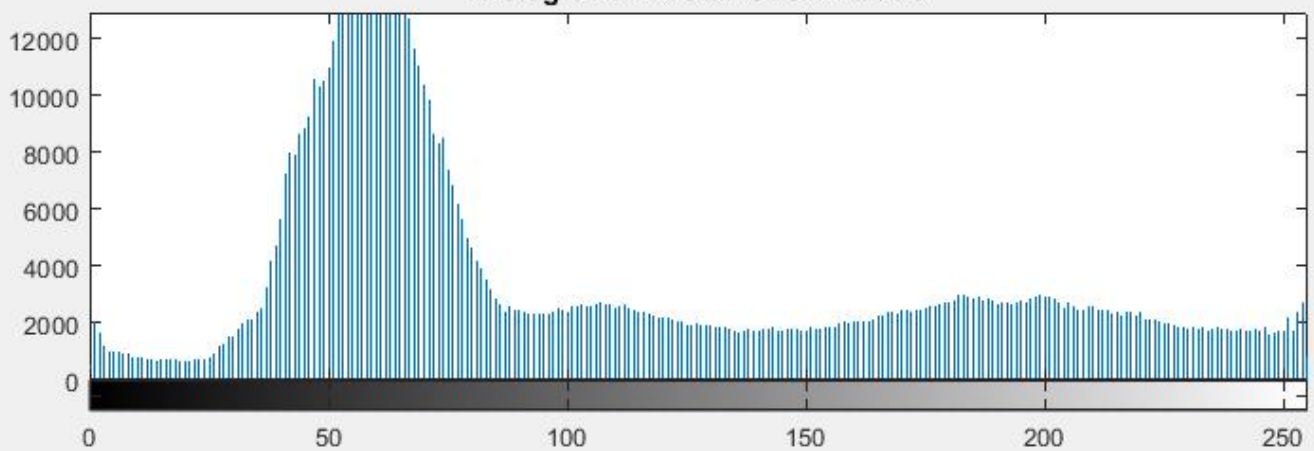




**Arco en Gris Invertido**



**Histograma Arco en Gris Invertido**



#### **Conclusiones:**

Como se muestran en los histogramas de cada imagen en blanco y negro, uno es el relejo del otro, pues ambos en su respectivo histograma, poseen la misma concentración de tonalidades de blanco y negro, pero reflejados.

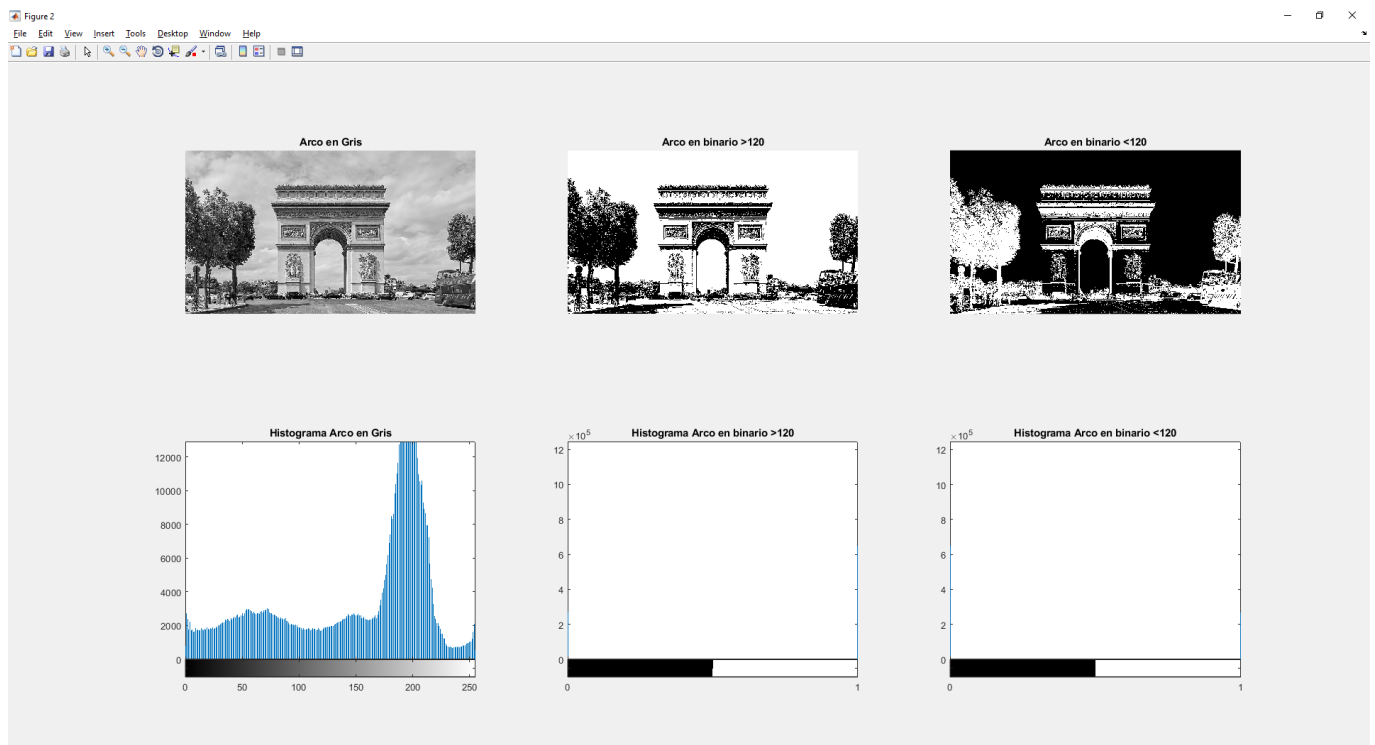
La siguiente imagen muestra el código para realizar la pregunta 4:

```
%Parte 4
p_umbral=120;
imagen_gray_binaria_positivo = imagen_gray>p_umbral;
imagen_gray_binaria_negativo = imagen_gray<p_umbral;

figure(2)
subplot(2,3,1), imshow(imagen_gray),
subplot(2,3,2), imshow(imagen_gray_binaria_positivo),
subplot(2,3,3), imshow(imagen_gray_binaria_negativo),
subplot(2,3,4), imhist(imagen_gray),
subplot(2,3,5), imhist(imagen_gray_binaria_positivo),
subplot(2,3,6), imhist(imagen_gray_binaria_negativo),

title('Arco en Gris')
title('Arco en binario >120')
title('Arco en binario <120')
title('Histograma Arco en Gris')
title('Histograma Arco en binario >120')
title('Histograma Arco en binario <120')
```

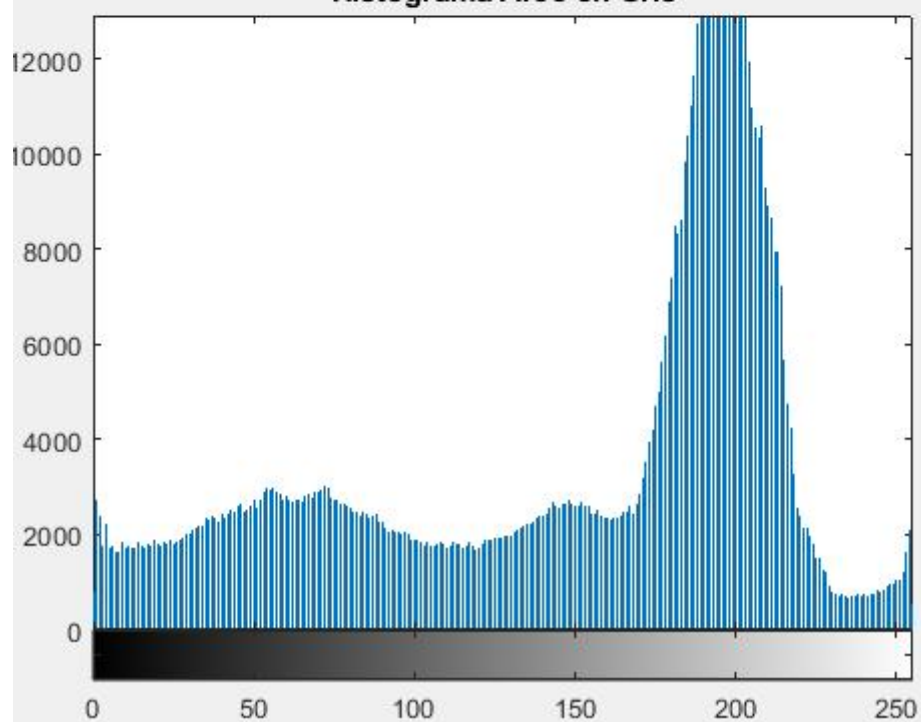
Y esto fue lo que conseguimos:



Arco en Gris



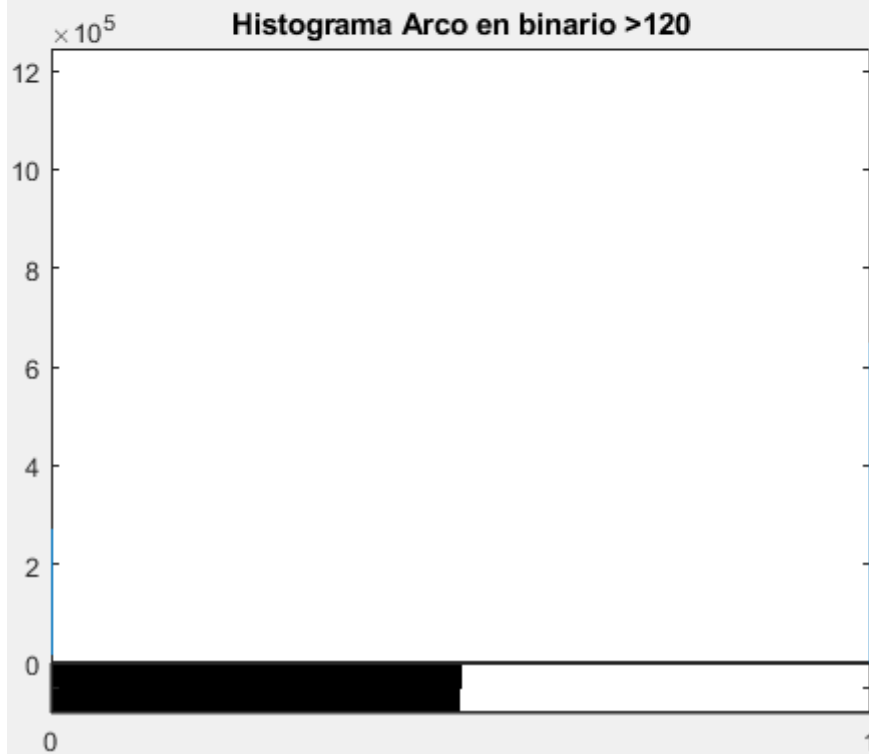
Histograma Arco en Gris



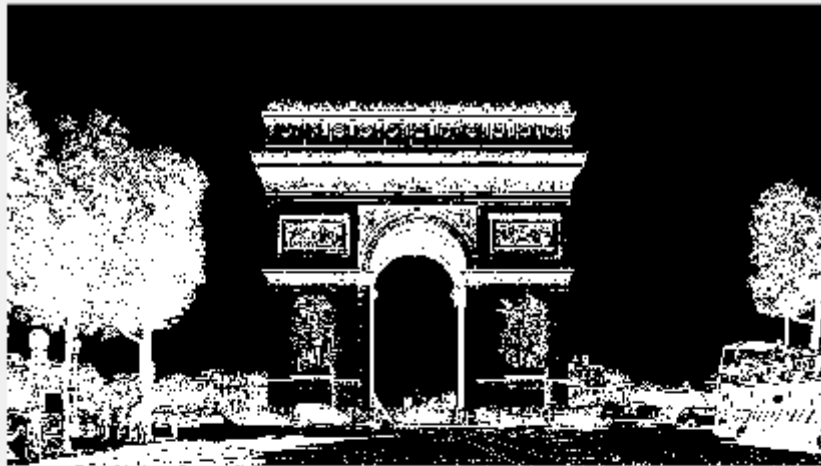
Arco en binario >120



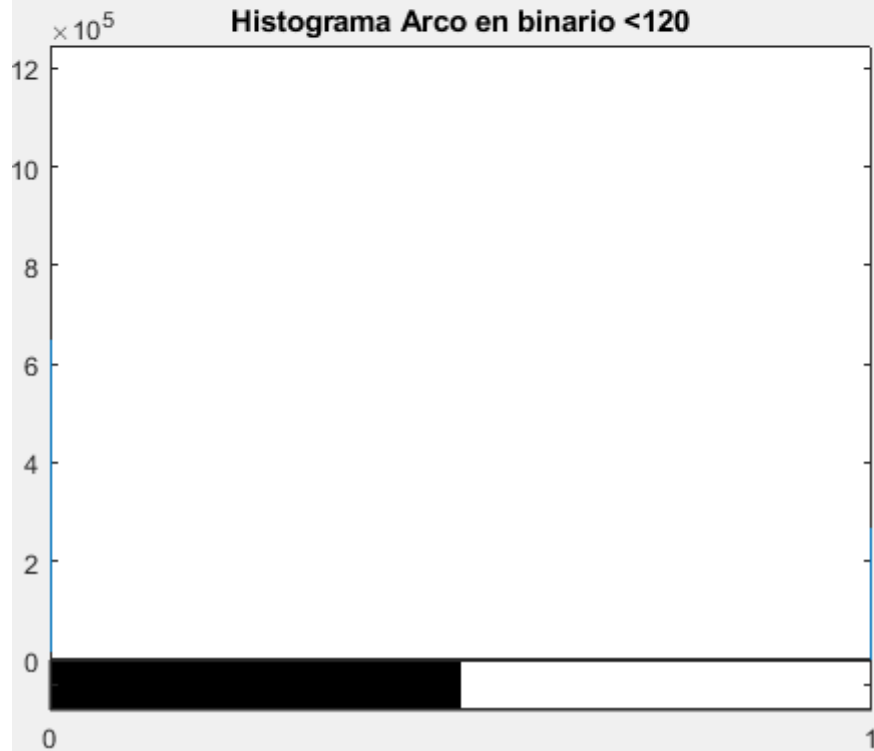
Histograma Arco en binario >120



Arco en binario <120



Histograma Arco en binario <120



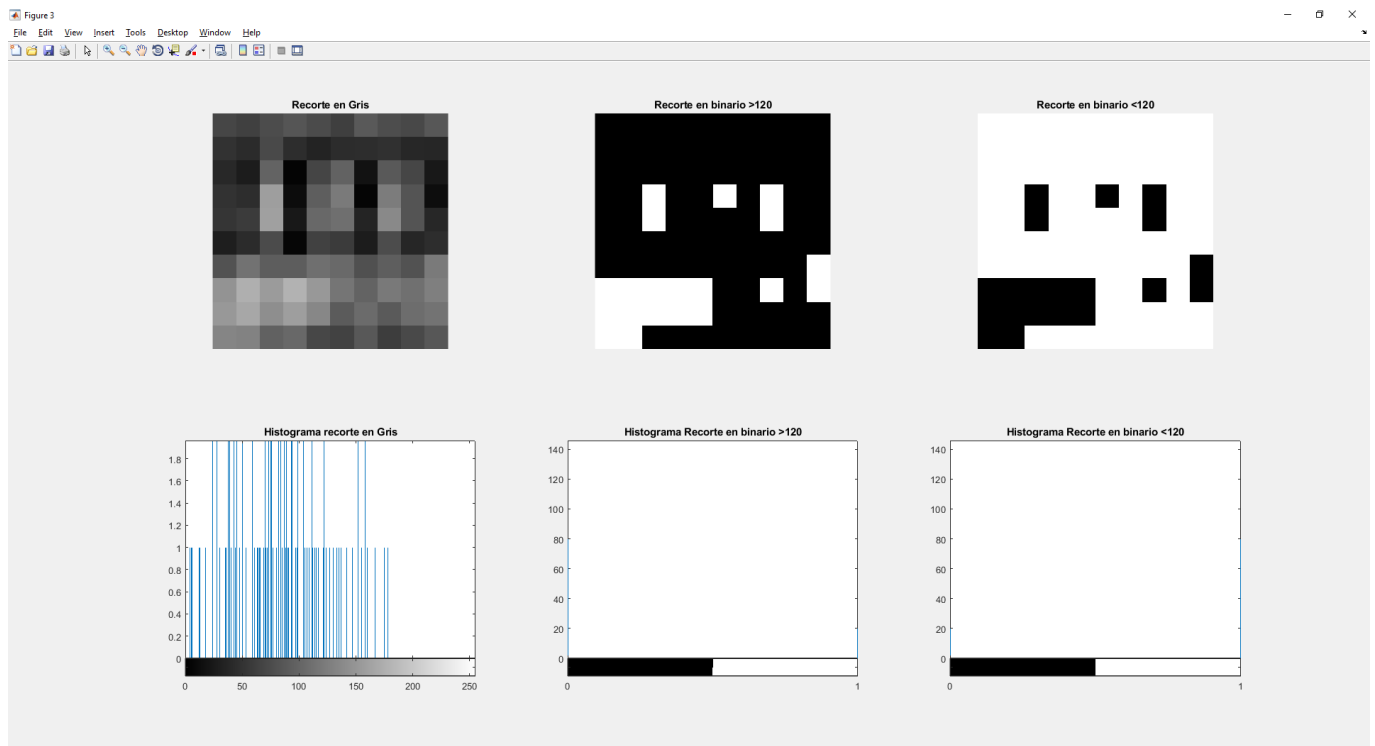


La siguiente imagen muestra el código para realizar la pregunta 5:

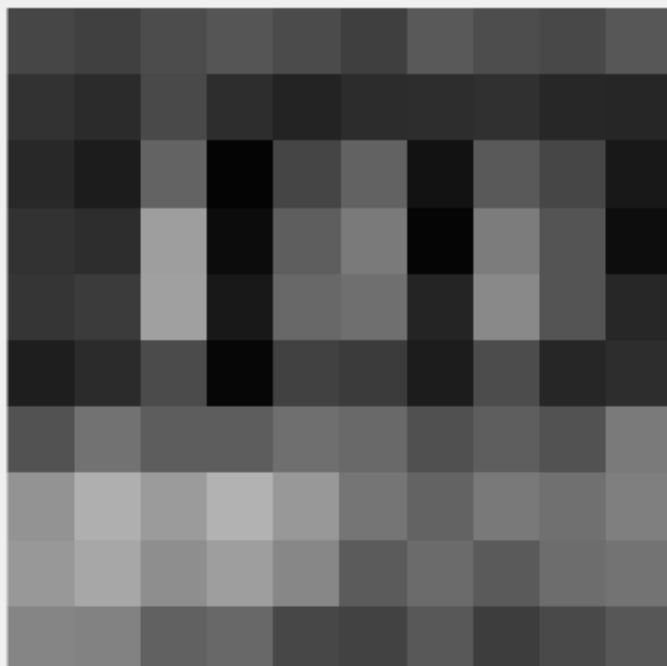
```
%Parte 5
recorte_10x10_gray = imcrop(imagen_gray,[781 178 9 9]);
recorte_10x10_bin_pos = imcrop(imagen_gray_binaria_positivo,[781 178 9 9]);
recorte_10x10_bin_nev = imcrop(imagen_gray_binaria_negativo,[781 178 9 9]);

figure(3)
subplot(2,3,1), imshow(recorte_10x10_gray), title('Recorte en Gris')
subplot(2,3,2), imshow(recorte_10x10_bin_pos), title('Recorte en binario >120')
subplot(2,3,3), imshow(recorte_10x10_bin_nev), title('Recorte en binario <120')
subplot(2,3,4), imhist(recorte_10x10_gray), title('Histograma recorte en Gris')
subplot(2,3,5), imhist(recorte_10x10_bin_pos), title('Histograma Recorte en binario >120')
subplot(2,3,6), imhist(recorte_10x10_bin_nev), title('Histograma Recorte en binario <120')
```

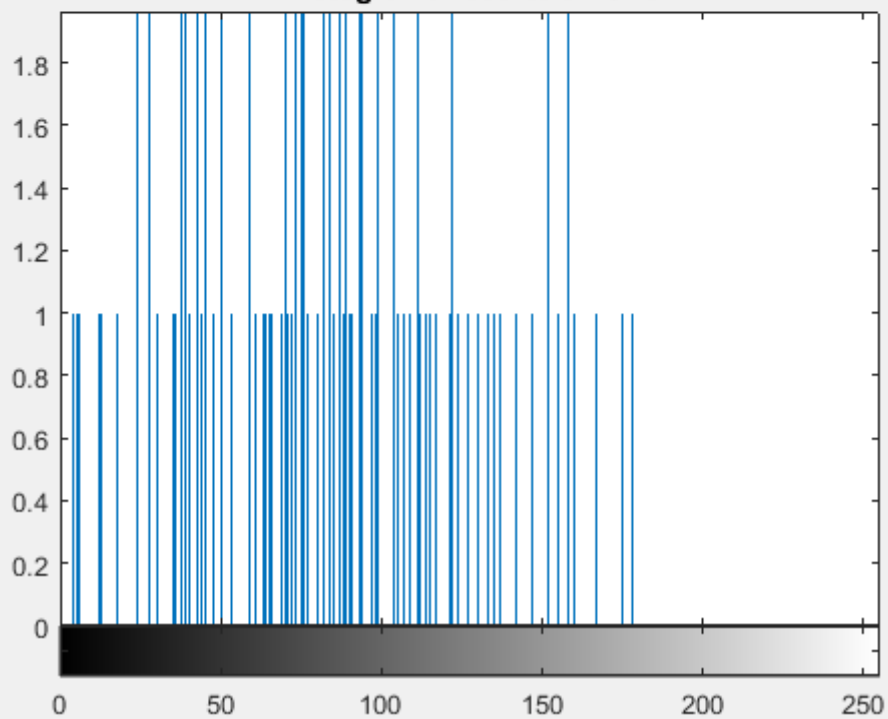
Y esto fue lo que conseguimos:



**Recorte en Gris**



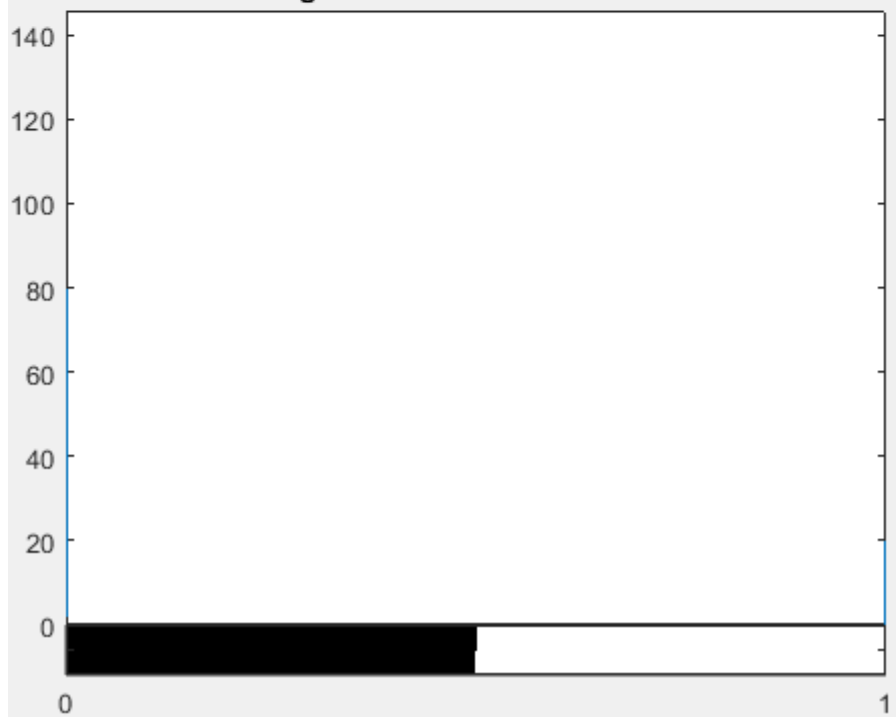
**Histograma recorte en Gris**



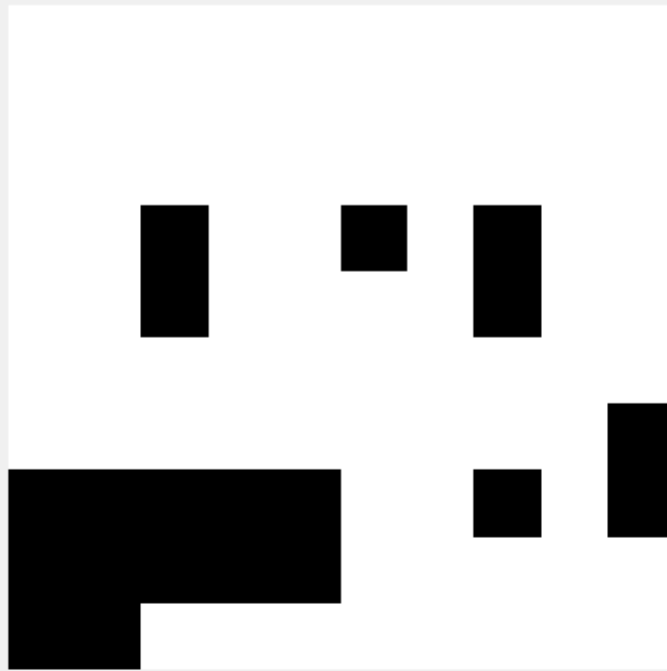
**Recorte en binario >120**



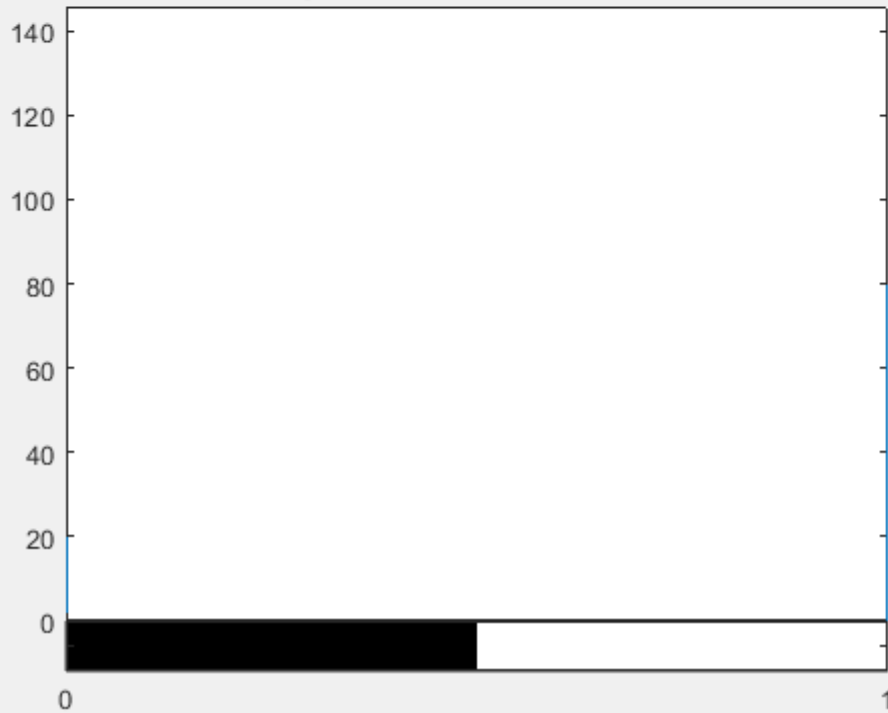
**Histograma Recorte en binario >120**



**Recorte en binario <120**



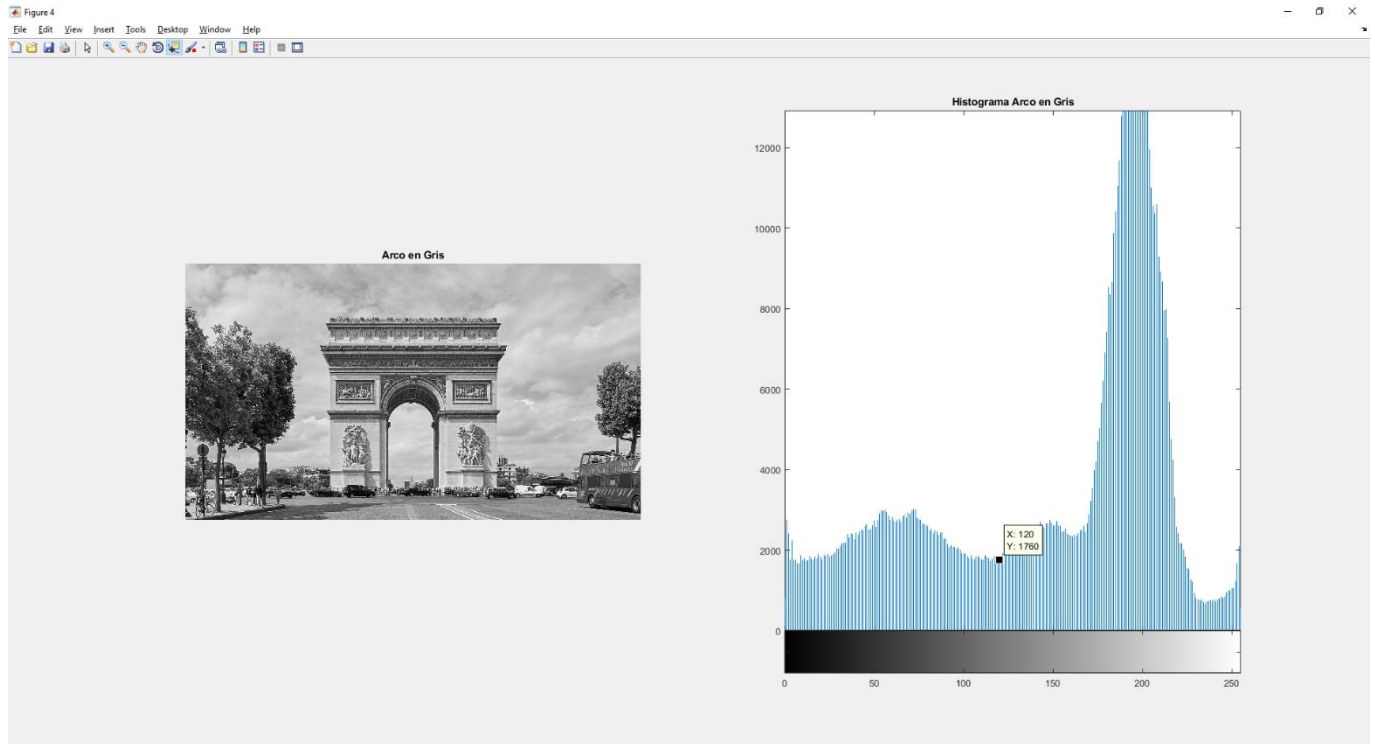
**Histograma Recorte en binario <120**



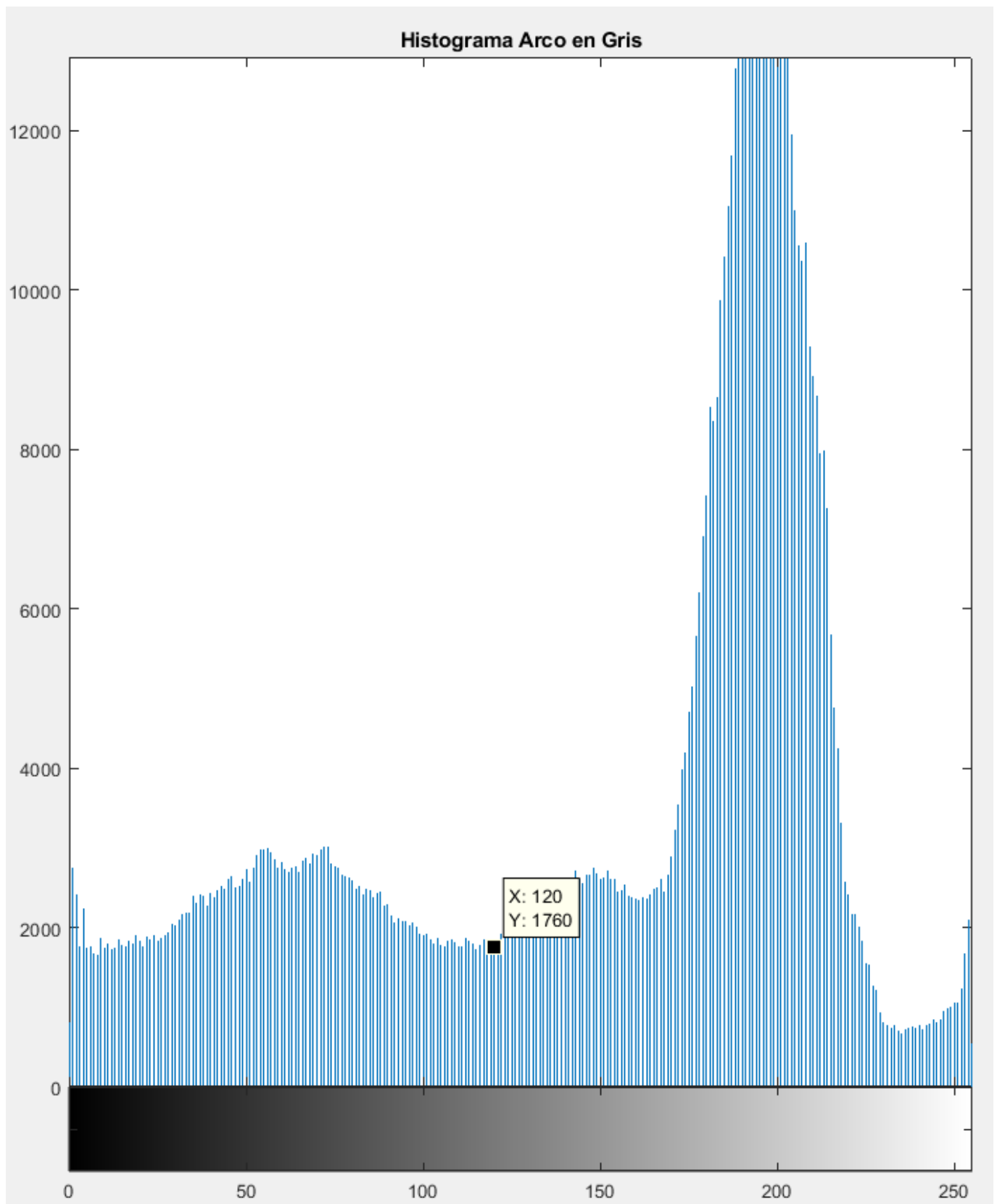
La siguiente imagen muestra el código para realizar la pregunta 6:

```
$Parte 6
figure(4)
subplot(1,2,1), imshow(imagen_gray), title('Arco en Gris')
subplot(1,2,2), imhist(imagen_gray), title('Histograma Arco en Gris')
```

Y esto fue lo que conseguimos:







### Conclusión:

Si tomamos el punto 120 como la mitad del histograma, podemos deducir que hay muchas más tonalidades de blanco que de negras en la imagen. Además, si polarizamos la imagen en ese punto esta se volverá mucho mas clara si ponemos el umbral mayor a 120, caso contrario, se oscurecerá.

La siguiente imagen muestra el código para realizar la pregunta 7:

```
%Punto 7
p1=50
p2=150
aumbra111=agray>p1;
aumbra112=agray<p1;
aumbra121=agray>p2;
aumbra122=agray<p2;

figure(5)
subplot(2,3,1)
imshow(agray)
subplot(2,3,2)
imshow(aumbra111),title('Umbral p1>50')
subplot(2,3,3)
imshow(aumbra112),title('Umbral p1<50')
subplot(2,3,4)
imshow(agray)
subplot(2,3,5)
imshow(aumbra121),title('Umbral p2>120')
subplot(2,3,6)
imshow(aumbra122),title('Umbral p2<120')
```

Y esto fue lo que conseguimos:



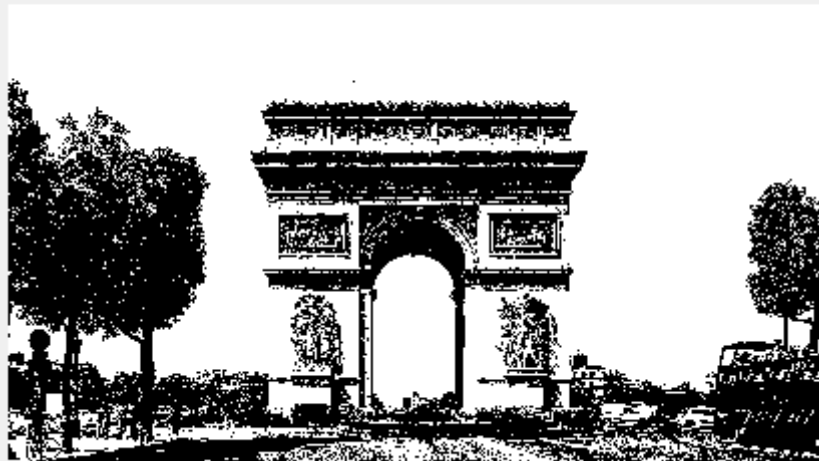
Umbral  $p1 > 50$

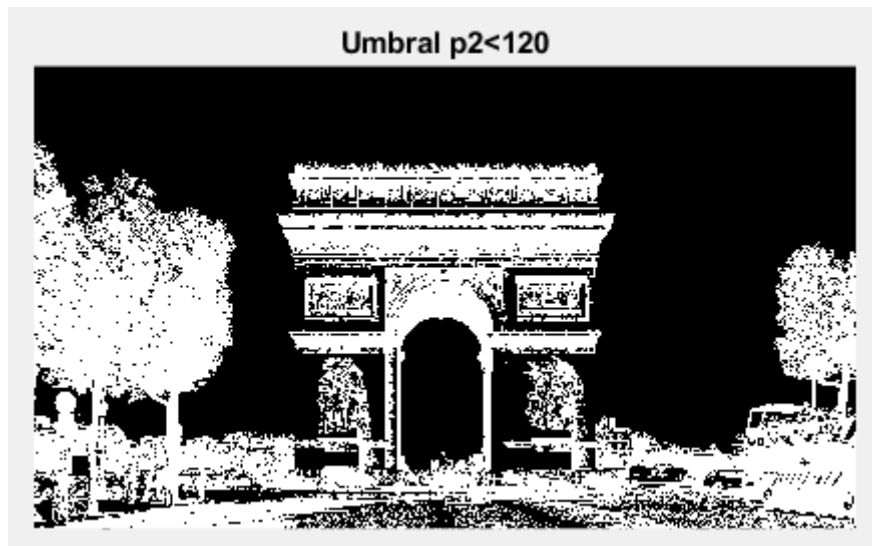


Umbral  $p1 < 50$



Umbral  $p2 > 120$





Como se observa en las imágenes, mientras más cercano es el umbral a cero y se usa la comparación “menor que” la imagen se oscurece, aumentando la tonalidad de negros en la imagen. Pero, en el caso de “mayor que” la imagen se aclara, es decir, aumenta la tonalidad de blancos.

## 7. ENTREGABLES

Trabajando con esta imagen



Se realizó el siguiente código para poder aislar el azul mediante un valor de umbral simple y un valor de umbral por intervalos.

```
clc, clear all, close all

imagen = imread('celulares2.jpg');
imagen_blue = imagen(:,:,3);

figure(1)
subplot(1,2,1), imshow(imagen_blue), title('Capa azul')
subplot(1,2,2), imhist(imagen_blue), title('Histograma de la Capa azul')

p_umbral = 120;
imagen_blue_aislado = imagen_blue < p_umbral;

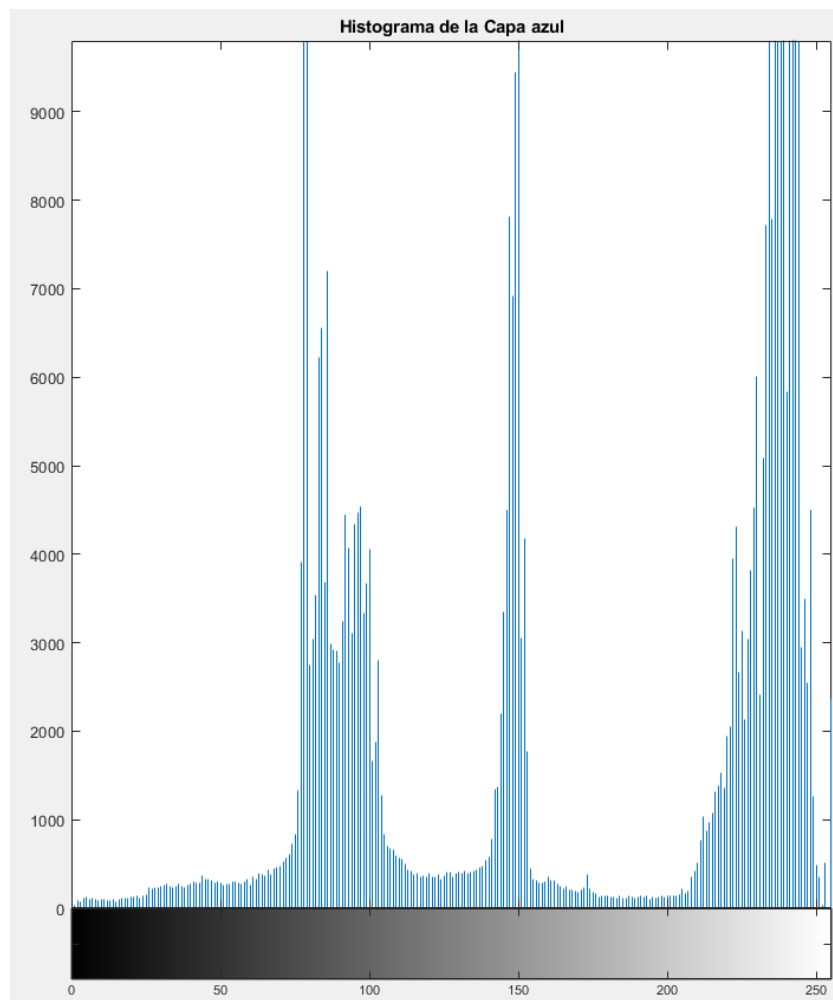
figure(2)
subplot(1,2,1), imshow(imagen_blue), title('Imagen completa')
subplot(1,2,2), imshow(imagen_blue_aislado), title('Azul aislado')

p_sup = 157;
p_inf = 130;
imagen_blue_aislado_intervalos = imagen_blue > p_inf & imagen_blue < p_sup;

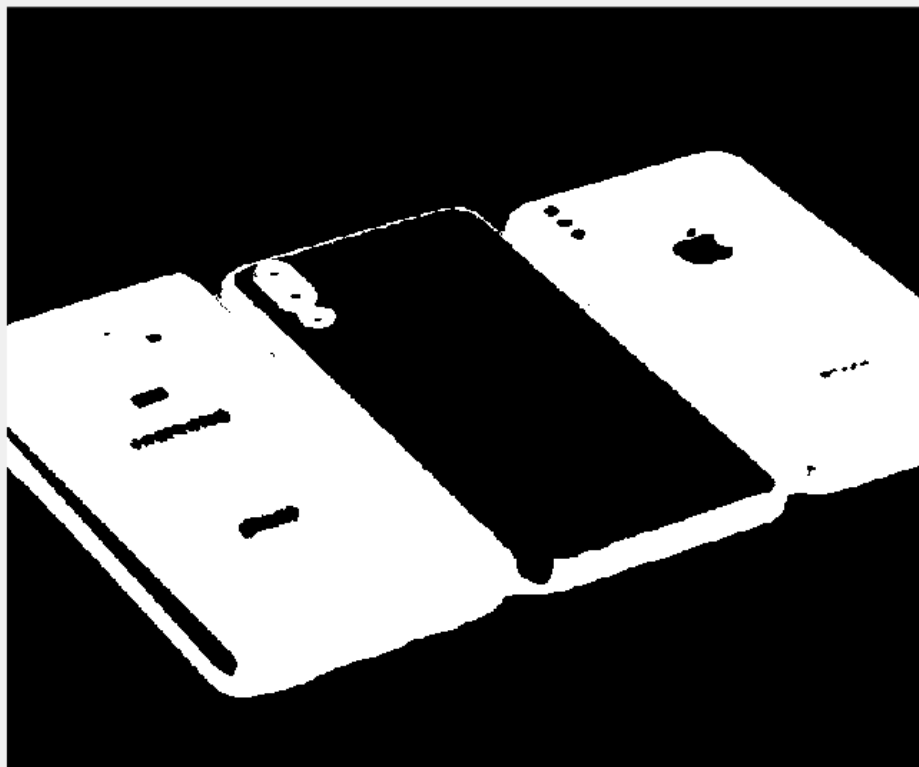
figure(3)
subplot(1,2,1), imshow(imagen_blue), title('Imagen completa')
subplot(1,2,2), imshow(imagen_blue_aislado_intervalos), title('Azul aislado en intervalos')
```



## Obteniendo



Azul aislado



Azul aislado en intervalos

