

Procesamiento Digital de Imágenes (PDI)

Universidad Nacional de Asunción

M. Sc. José Luis Vázquez Noguera

jl vazquez@pol.una.py

1era Clase de Laboratorio

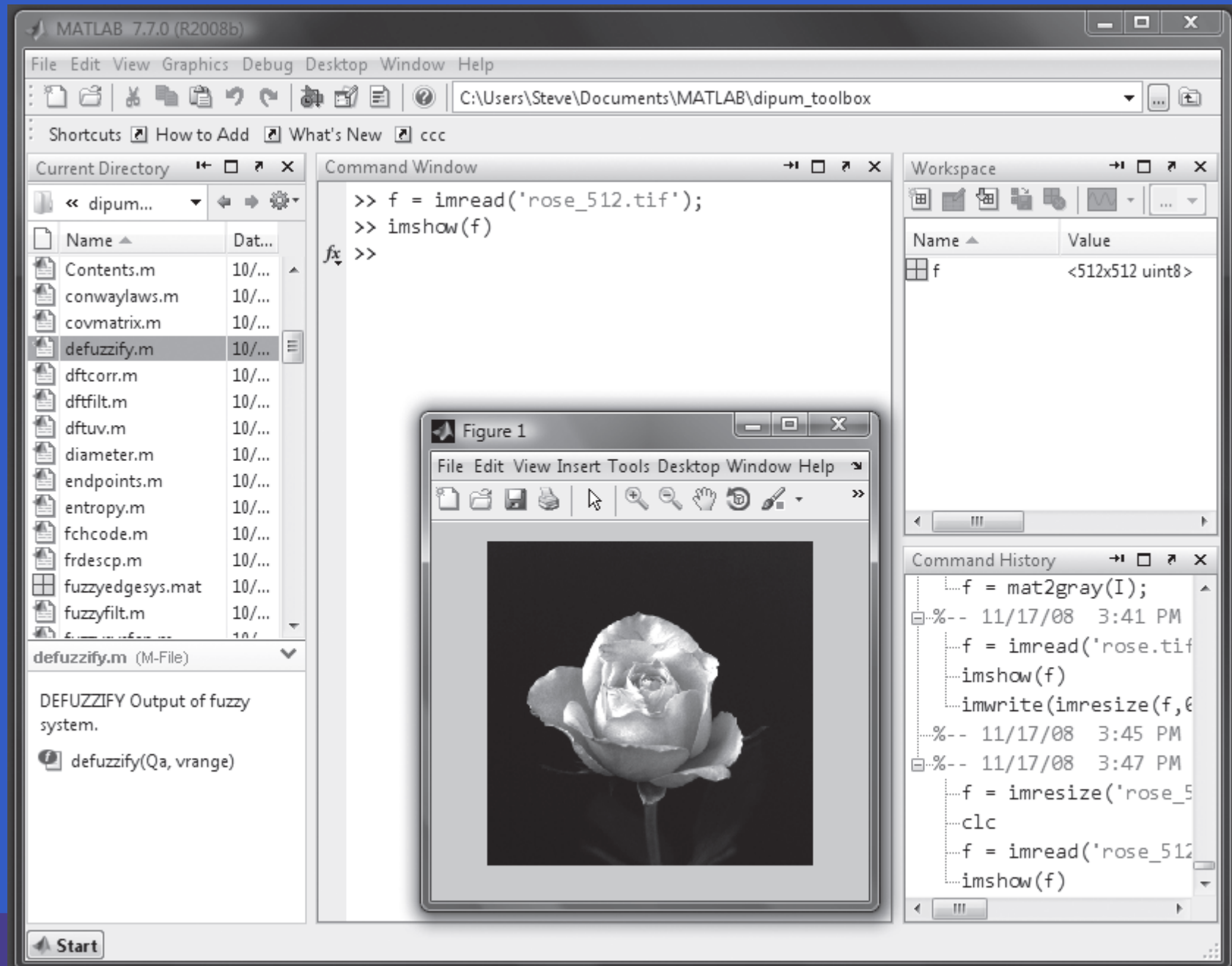
Capítulo 1

Ambiente de Trabajo Matlab

Contenido

- Escritorio de Matlab
- Editor de Matlab para crear archivos M
- Obtener ayuda

Escritorio de Matlab



Editor Matlab

El editor de MATLAB es una de las más importante herramientas de escritorio. Su objetivo principal es crear y editar archivos. Estos archivos se llaman los archivos M debido usan la extensión. m.

Obtener Ayuda

La principal forma de obtener ayuda es usar el navegador de ayuda de MATLAB, o bien haciendo clic en el signo de interrogación (?)

Capítulo 2

Fundamentos

Contenido

- Representación Digital de Imágenes
- Lectura de imágenes
- Visualización de Imágenes
- Escritura de imágenes
- Clases de Datos
- Tipos de imágenes

Representación de una Imagen Digital

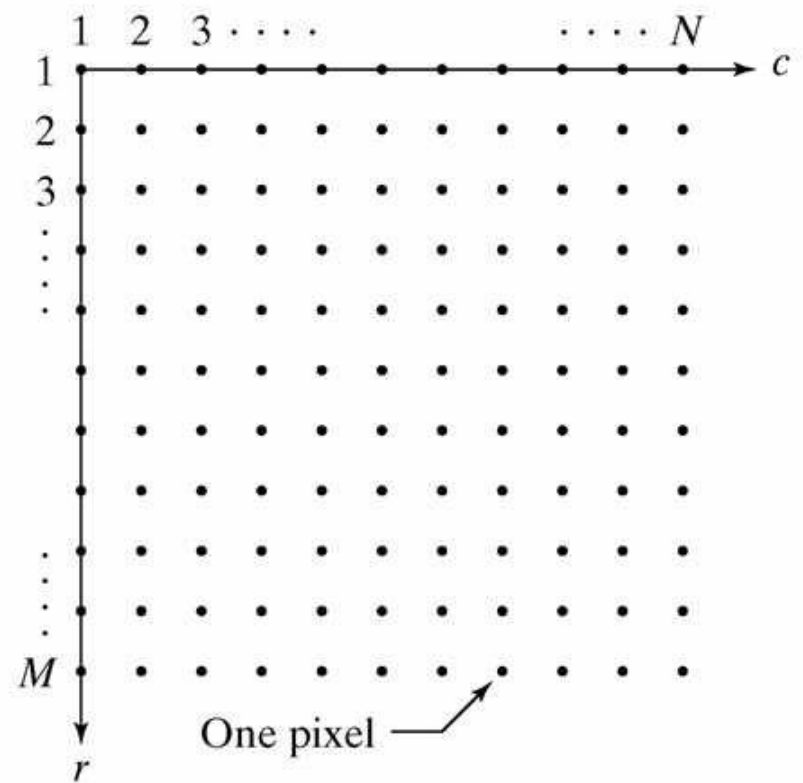
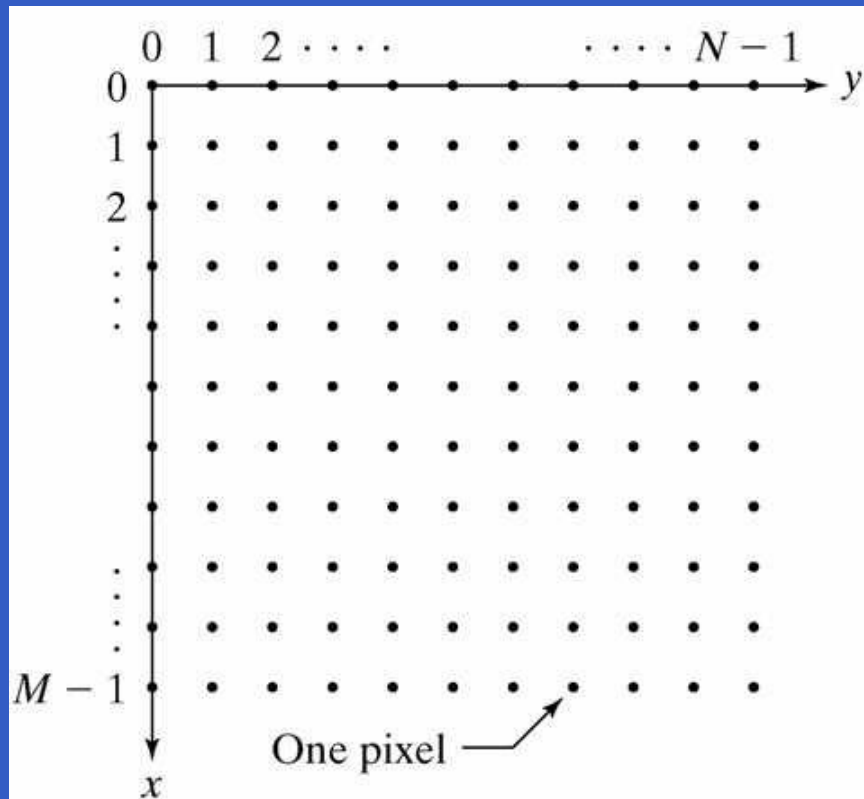
Una imagen puede ser definida como una función de dos dimensiones

$$f(x,y),$$

donde x e y son coordenadas espaciales, y f es la intensidad de la imagen f en el punto (x,y) .

Cuando x , y e los valores de la amplitud de la función f son cantidades discretas finitas, a dicha imagen se le llama imagen digital.

Convención



En muchos libros de PDI

En el Toolbox de PDI de Matlab

Imagen como Matrices

Una imagen puede ser representada como una matriz

$$f = \begin{bmatrix} f(1, 1) & f(1, 2) & \cdots & f(1, N) \\ f(2, 1) & f(2, 2) & \cdots & f(2, N) \\ \vdots & \vdots & \ddots & \vdots \\ f(M, 1) & f(M, 2) & \cdots & f(M, N) \end{bmatrix}$$

Lectura de imágenes

```
imread('filename')
```

Some examples:

- `f=imread('chestxray.jpg');`
- `f=imread('D:\myimages\chestxray.jpg');`
- `f=imread('.\myimages\chestxray.jpg');`

Formato de imágenes soportados

Format Name	Description	Recognized Extensions
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photographic Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Función size

```
size(imagematrix)
```

```
>> size(f)
```

```
ans =
```

```
    494    600
```

```
>> [M,N]=size(f);
```

```
>> whos f
```

Name	Size	Bytes	Class
f	494x600	296400	uint8 array

Visualización de imágenes

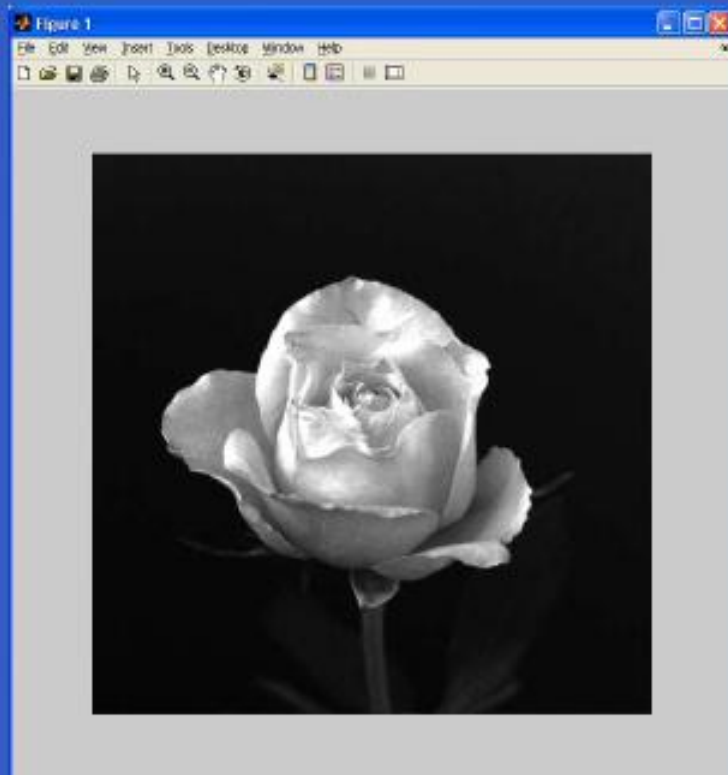
```
imshow(f, G)
```

- `imshow(f, [low, high])`

Se muestran color negro todos los valores menores o igual a low, y blanco todos los valores mayores o igual a high

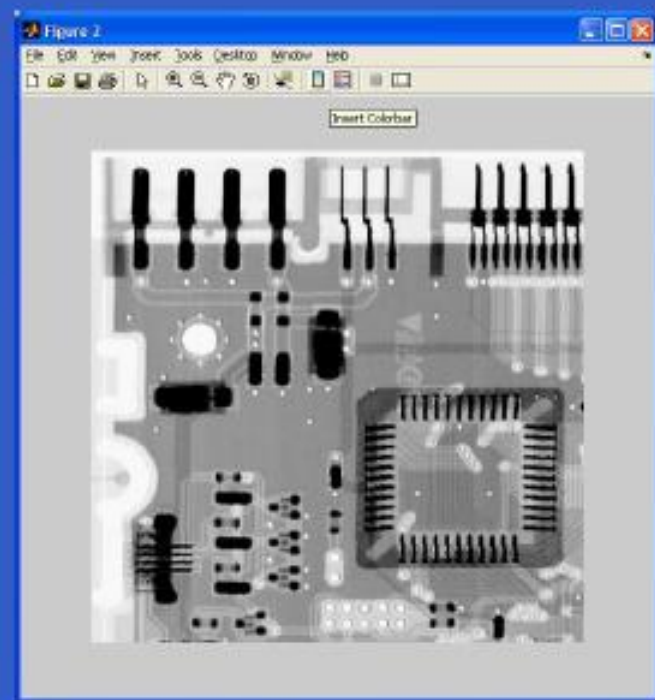
Visualización de imágenes

```
>> f=imread('rose_512.tif');  
>> imshow(f)
```



Visualización de imágenes

```
>> f=imread('rose_512.tif');  
>> g=imread('cktboard.tif');  
>> imshow(f), figure, imshow(g)
```



Escritura de imágenes

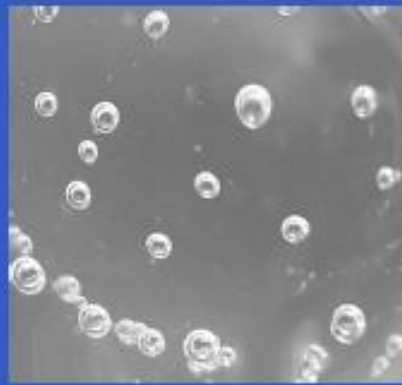
```
imwrite(f, 'filename')
```

- `imwrite(f, 'patient10_run1', 'tif')`
- `imwrite(f, 'patient10_run1.tif')`

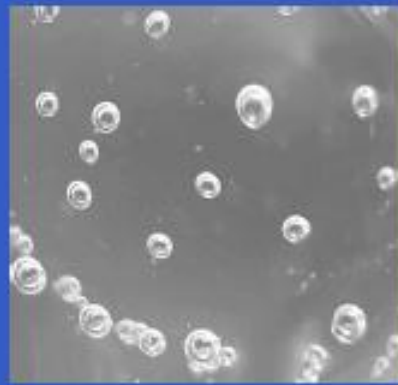
```
imwrite(f, 'filename.jpg', 'quality', q)
```

Donde q es un número entero entre 0 y 100 cuanto menor sea el número mayor será la degradación debido a la compresión JPEG

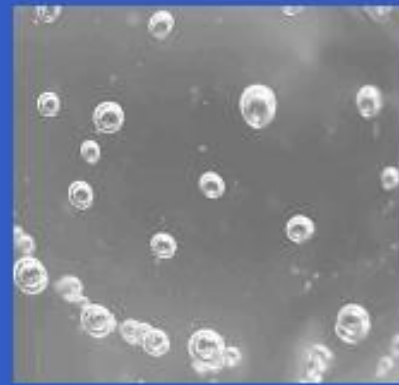
Escritura de imágenes



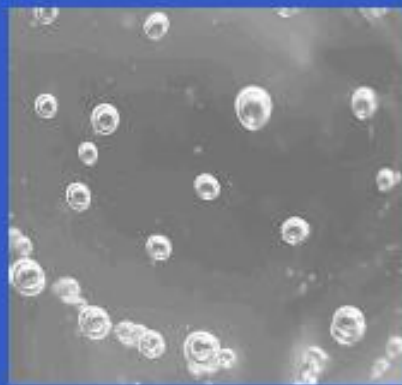
$q = 100$



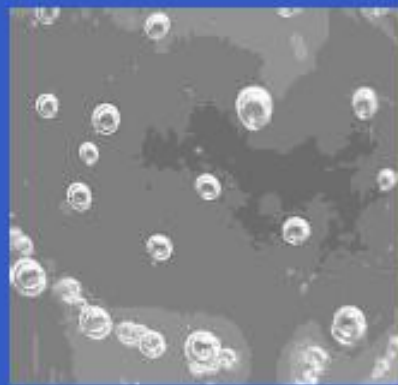
$q = 50$



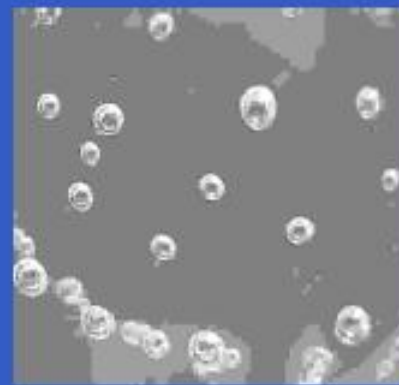
$q = 25$



$q = 15$



$q = 5$



$q = 0$

Escritura de imágenes

```
imfinfo filename
```

```
>> imfinfo bubbles25.jpg
```

```
ans =
```

```
      Filename: 'bubbles25.jpg'
    FileModDate: '02-Feb-2005 09:34:50'
      FileSize: 13354
        Format: 'jpg'
FormatVersion: ''
        Width: 720
        Height: 688
      BitDepth: 8
     ColorType: 'grayscale'
FormatSignature: ''
NumberOfSamples: 1
   CodingMethod: 'Huffman'
  CodingProcess: 'Sequential'
        Comment: {}
```

Escritura de imágenes

```
K=imfinfo('bubbles25.jpg');
```

```
image_bytes=K.Width*K.Height*K.BitDepth/8;
```

```
compresion_bytes=K.FileSize;
```

```
compresion_tasa=image_bytes/compresion_bytes
```

```
compresion_tasa =
```

```
38.2370
```

Clases de Datos

Name	Description
double	Double-precision, floating-point numbers in the approximate range -10^{308} to 10^{308} (8 bytes per element).
uint8	Unsigned 8-bit integers in the range [0,255] (1 byte per element).
uint16	Unsigned 16-bit integers in the range [0,65535] (2 bytes per element).
uint32	Unsigned 32-bit integers in the range [0,4294967295] (4 bytes per element).
int8	Signed 8-bit integers in the range [-128,127] (1 byte per element).
int16	Signed 16-bit integers in the range [-32768,32767] (2 bytes per element).
int32	Signed 32-bit integers in the range [-2147483648,2147483647] (4 bytes per element).
single	Single-precision floating-point numbers with values in the approximate range -10^{38} to 10^{38} (4 bytes per element).
char	Characters (2 bytes per element).
logical	Values are 0 or 1 (1 byte per element).

Tipos de imágenes

- Imágenes Binarias
- Imágenes en escala de grises
- Imágenes RGB

Imágenes Binarias

Dado que hay sólo dos valores posibles para cada píxel, solo necesitamos un bit por píxel. Una imagen binaria es un arreglo lógico de 0s y 1s

Un arreglo numérico es convertido a binario usando la función `logical`

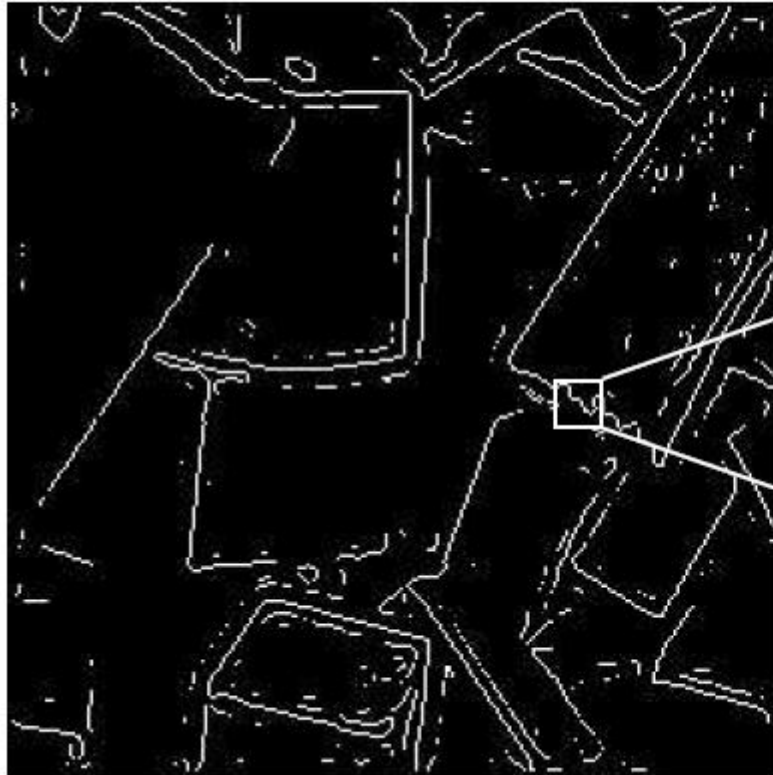
```
B=logical(A)
```

Para testear si un arreglo es lógico debemos usar la función `islogical`

```
islogical(C)
```

Si C es un arreglo lógico, esta función retorna un 1. En otro caso retorna un 0.

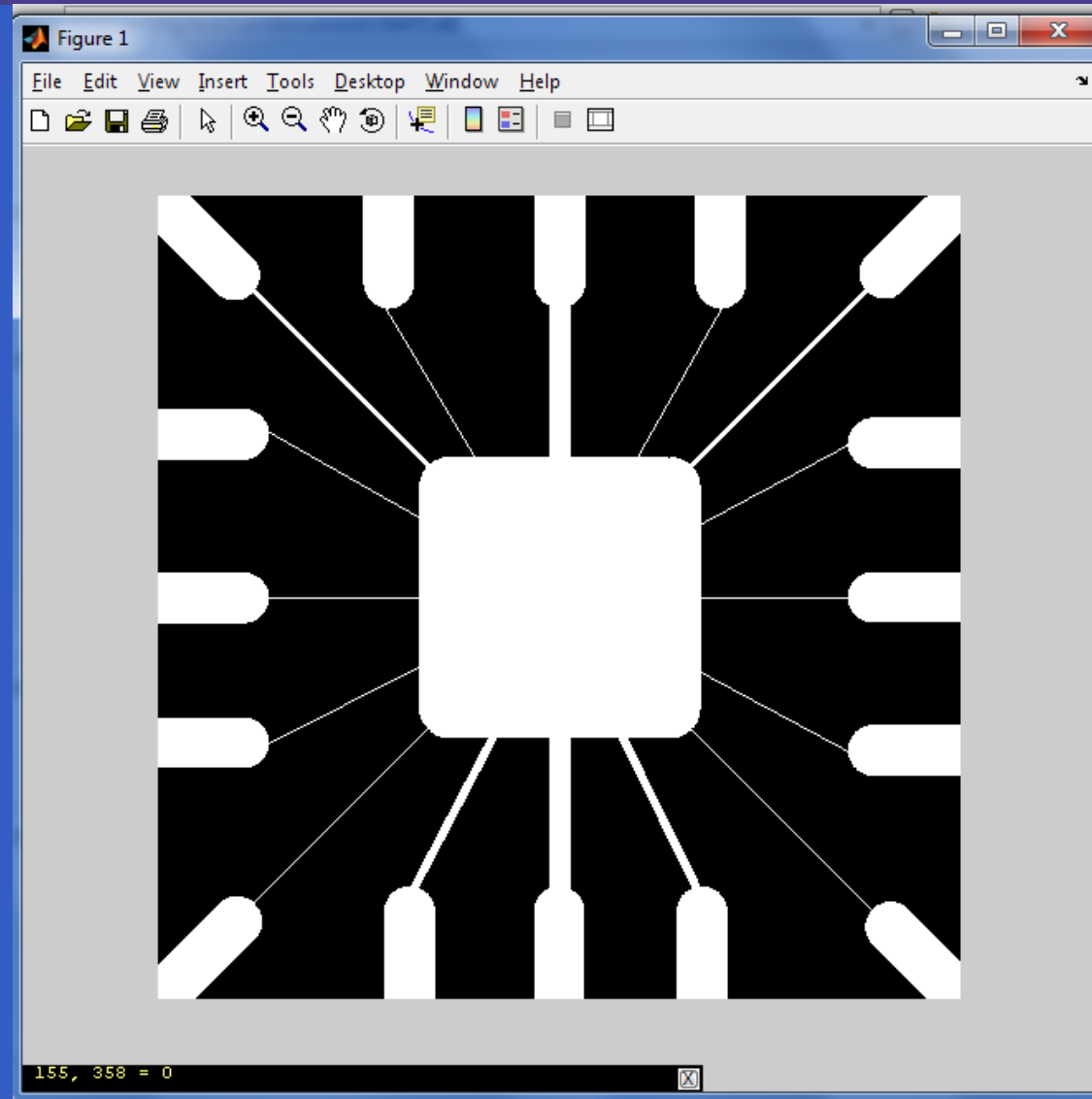
Imágenes Binarias



1	1	0	0	0	0
0	0	1	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	0	0	0	1

Imágenes Binarias

```
f = imread('binary.tif');  
figure,imshow(f),pixval on
```



Imágenes en Escala de Grises

Cada pixel es un nivel de gris normalmente de 0 (negro) a 255 (blanco) cuándo la imagen es de clase uint8, o de 0 (negro) a 65535 (blanco) uint16. Si la imagen es de clase double los valores son números en puntos flotantes en la escala [0 1]

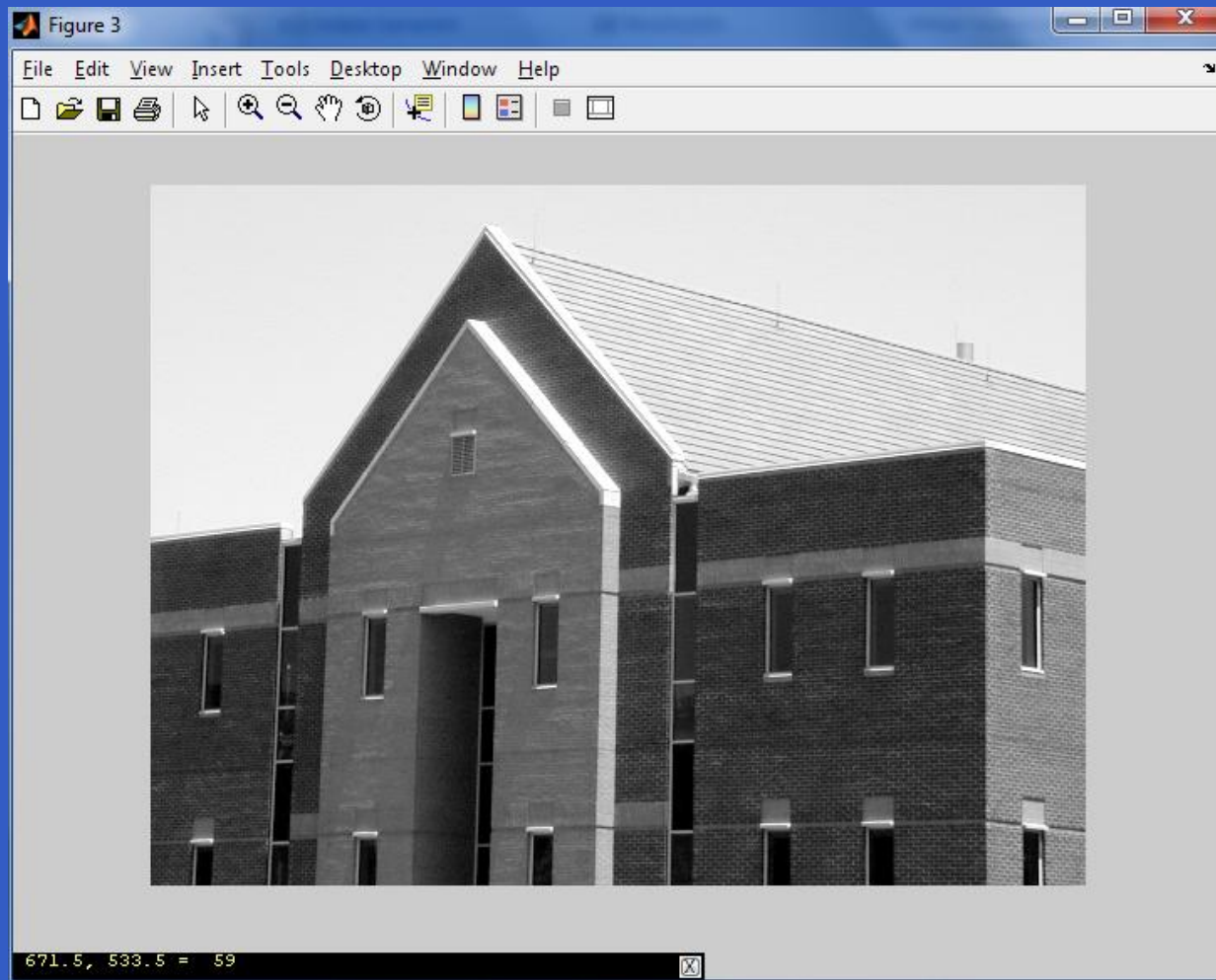
Imágenes en Escala de Grises



230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Imágenes en Escala de Grises

```
f = imread('gray.tif');  
figure,imshow(f),pixval
```



Imágenes RGB

Aquí cada pixel tiene un color en particular; que el color que se describe por el cantidad de rojo, verde y azul. Si cada uno de estos componentes tiene un rango de 0 a 255. Esto da un total de $255^3 = 16.777.216$ diferentes posibles colores en la imagen

Imágenes RGB



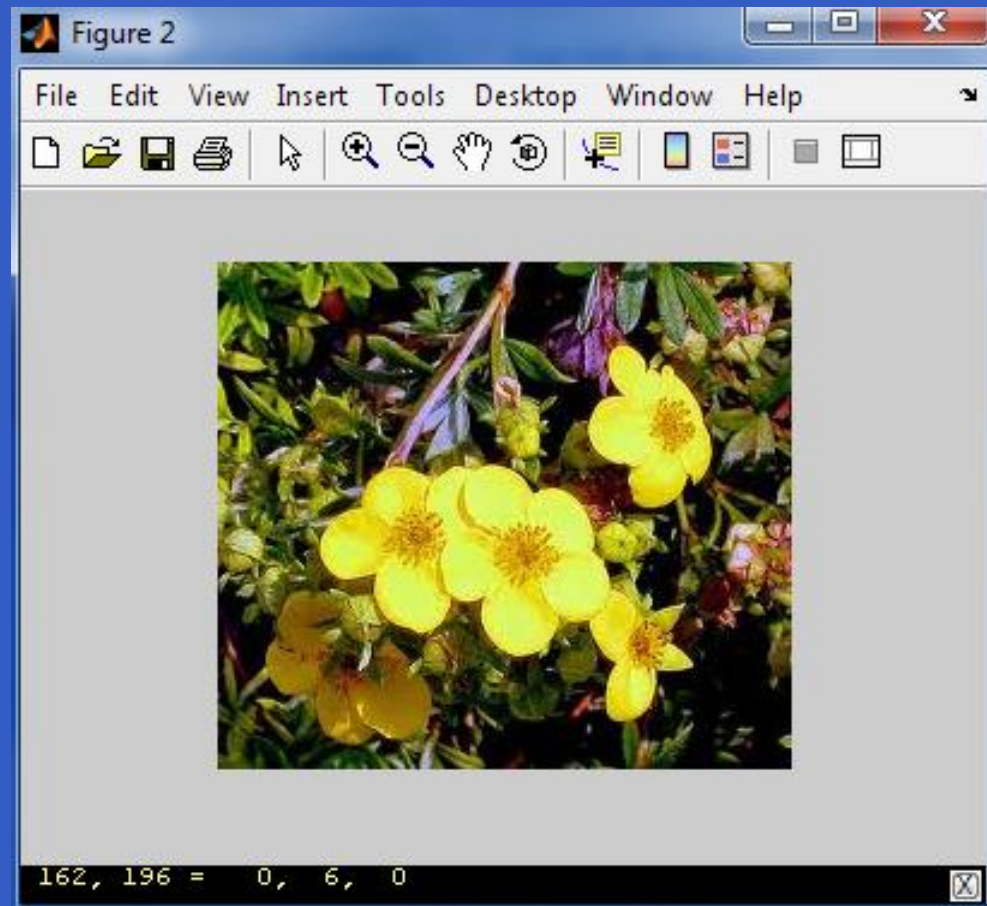
49	55	56	57	52	53
58	60	60	58	55	57
58	58	54	53	55	56
83	78	72	69	68	69
88	91	91	84	83	82
69	76	83	78	76	75
61	69	73	78	76	76

64	76	82	79	78	78
93	93	91	91	86	86
88	82	88	90	88	89
125	119	113	108	111	110
137	136	132	128	126	120
105	108	114	114	118	113
96	103	112	108	111	107

66	80	77	80	87	77
81	93	96	99	86	85
83	83	91	94	92	88
135	128	126	112	107	106
141	129	129	117	115	101
95	99	109	108	112	109
84	93	107	101	105	102

Imágenes RGB

```
f = imread('rgb.tif');  
figure,imshow(f),paxval on
```



El espacio de color RGB

Se puede representar gráficamente como un cubo. Los vértices del cubo son los colores primarios (Rojo, Verde y Azul) y los colores secundarios (Cian, Magenta y Amarillo). Los niveles de gris podría situarse en la diagonal que une al color blanco con el negro.

