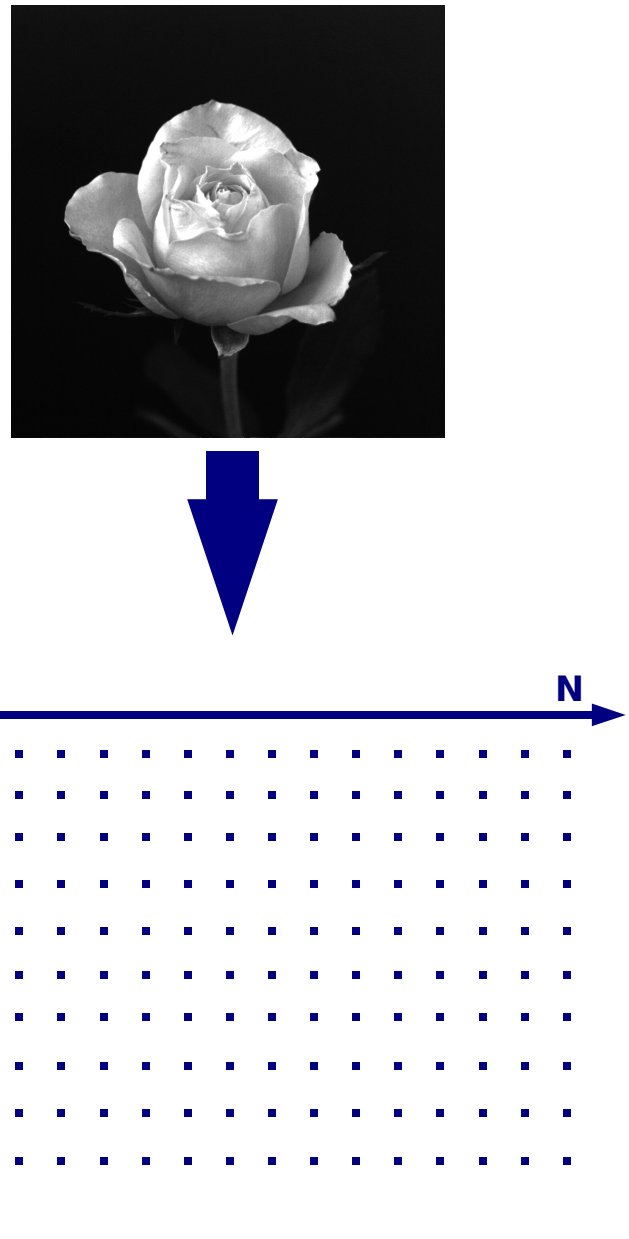
PRÁCTICA 1

MANEJO BÁSICO DE IMÁGENES CON MATLAB

David García Pérez

MANEJO BÁSICO DE IMÁGENES CON MATLAB

- Imagen tamaño NxM píxeles
- En Matlab: Imagen = Matriz/es
- Cada elemento de la matriz el tono de gris para imágenes en B/N
- Si la imagen es en color, existen 3 matrices, cada una representa el valor de un tono de color
- Los índices de la matriz son (r,c), donde r representa la fila (*row*) y c la columna (*column*)



MANEJO BÁSICO DE IMÁGENES CON MATLAB

Leyendo imágenes de disco:

```
>> f = imread('chestxray.jpg')
>> f = imread('D:\imagenes\chestxray.jpg')
>> f = imread('/home/user/chestxray.jpg')
```

Esto lo que hace es que para una imagen en blanco y negro, nos crea una matriz f, donde cada elemento tendrá un valor de nivel de gris:

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

MANEJO BÁSICO DE IMÁGENES CON MATLAB

Obteniendo información de f:

Para determinar el tamaño de la imagen:

```
>>size(f)
   ans =
        1024 1024
```

Si lo queremos almacenar en una variable:

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

Si lo que queremos es obtener información más detallada de la imagen:

```
>>whos f
  Name  Size          Bytes   Class
  f     249x500       373500  uint8 array
Grand total is 373500 elements using 373500 bytes
```

MANEJO BÁSICO DE IMÁGENES CON MATLAB

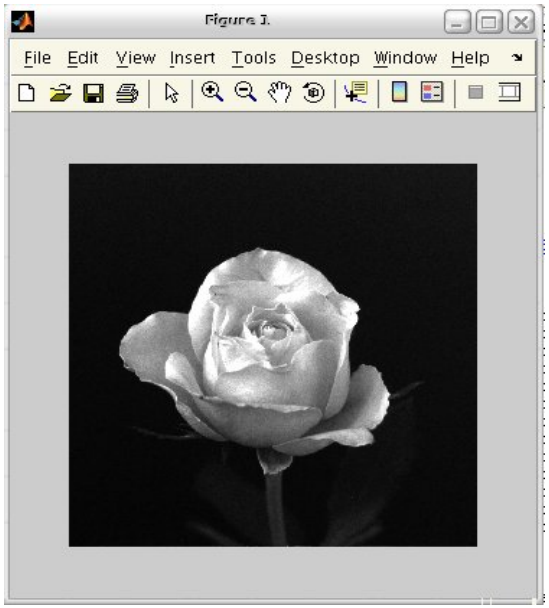
Mostrando Imágenes:

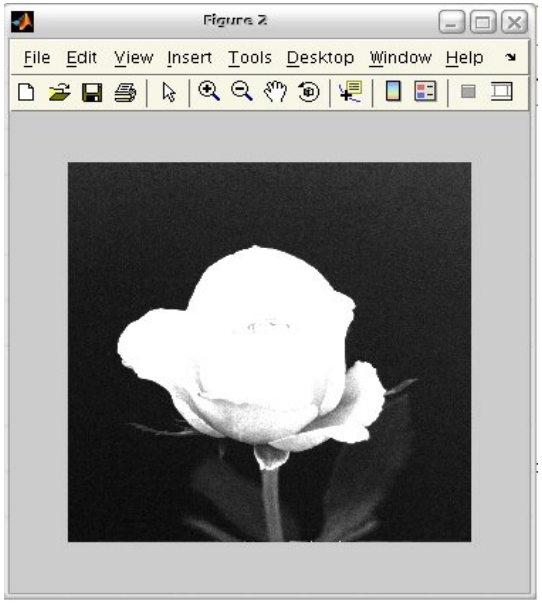
```
imshow(f,G)
```

donde f es la imagen a mostrar y G es el número de niveles de intensidad a mostrar. Si G se omite, se usa 256. La sintáxis es la siguiente

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

`imshow(f)`





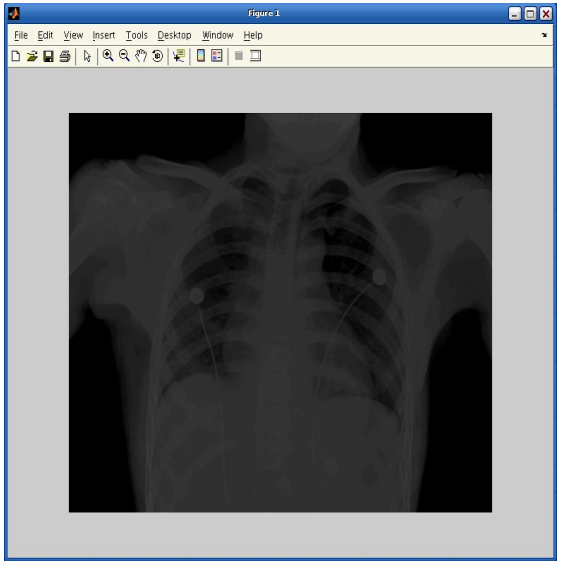
`imshow(f,[0 50])`

MANEJO BÁSICO DE IMÁGENES CON MATLAB

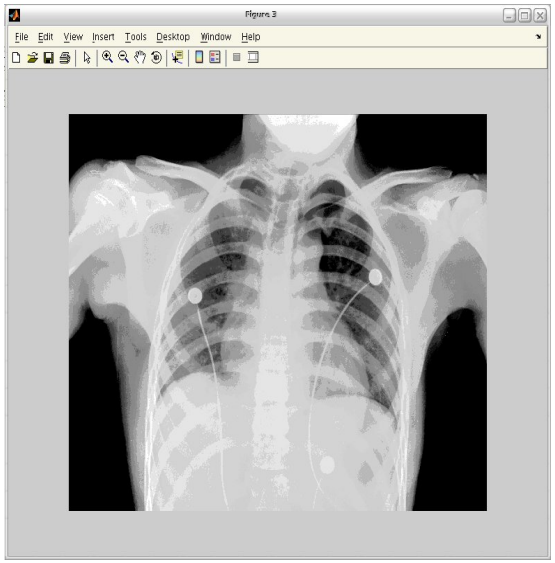
Una utilidad interesante de imshow es:

```
imshow(f,[])
```

esto lo que hace es expandir el rango dinámico de la imagen, pone como límite inferior, el valor mínimo de intensidad de la imagen y como límite superior su valor máximo



`imshow(f)`



`imshow(f,[])`

MANEJO BÁSICO DE IMÁGENES CON MATLAB

Dos apuntes más sobre `imshow(f)`:

`pixval`: Si se ejecuta `pixval` en la consola de Matlab, en la última figura abierta, se podrá ver el valor de intensidad cuando el ratón pasa sobre un píxel concreto. Y también se puede medir la distancia euclídea entre dos puntos.

Matlab usualmente cuando dibuja una nueva figura, sobreescribe la ventana de la figura anteriormente mostrada. Para mostrar la segunda figura en una ventana independiente, ejecutar lo siguiente

```
>> figure, imshow(f)
```

MANEJO BÁSICO DE IMÁGENES CON MATLAB

Guardando imágenes en disco:

```
>> imwrite(f, 'nombre_fichero')
```

donde f es la matriz que almacena la imagen y filename es el nombre de fichero donde vamos a guardar la imagen. filename tiene que tener una extensión que reconozca Matlab, o usar este otro formato:

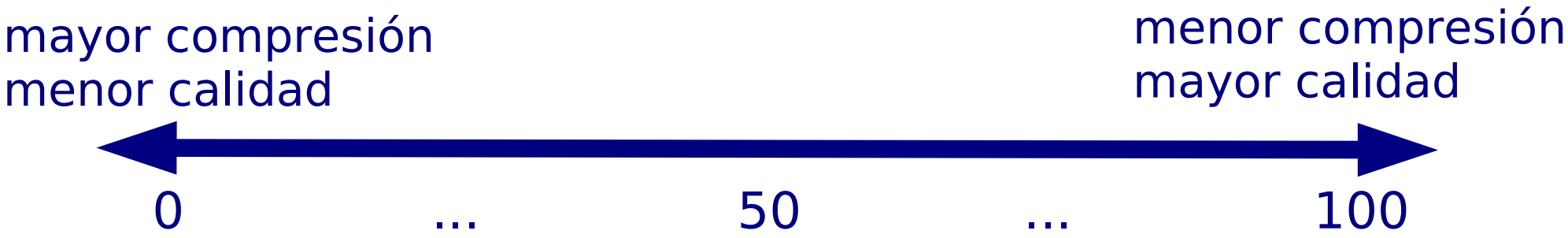
```
>> imwrite(f, 'nombre_fichero', 'tif)
```

En estas prácticas trabajaremos con formato tiff y/o jpeg. En el caso de jpeg también se puede especificar la calidad de la imagen:

```
>> imwrite(f, 'nombre.jpg', 'quality', q)
```

MANEJO BÁSICO DE IMÁGENES CON MATLAB

donde q es un numero de 0 a 100 que especifica la calidad de la imagen:



Obteniendo información de un fichero imagen almacenado en disco:

```
>> imfinfo nombre_fichero
```

MANEJO BÁSICO DE IMÁGENES CON MATLAB

```
>> imfinfo prueba.jpg
ans =
           Filename: 'prueba.jpg'
        FileModDate: '08-feb-2005
17:18:13'
           FileSize: 6125
             Format: 'jpg'
      FormatVersion: ''
              Width: 600
             Height: 494
           BitDepth: 8
          ColorType: 'grayscale'
    FormatSignature: ''
    NumberOfSamples: 1
       CodingMethod: 'Huffman'
      CodingProcess: 'Sequential'
            Comment: {}
```

MANEJO BÁSICO DE IMÁGENES CON MATLAB

Clases de imágenes:

double Doble precisión, números en punto flotante que varían en un rango aproximado de -10^{308} a 10^{308} (8 bytes por elemento)

uint8 Enteros de 8 bits en el rango de [0,255] (1 byte por elemento)

uint16 Enteros de 16 bits en el rango de [0, 65535] (2 bytes por elemento)

uint32 Enteros de 32 bits en el rango de [0, 4294967295] (4 bytes por elemento)

int8 Enteros de 8 bits en el rango de [-128, 127] (1 byte por elemento)

int16 Enteros de 16 bits en el rango de [-32768, 32767] (2 bytes por elemento)

int32 Enteros de 32 bits en el rango de [-2147483648,2147483647] (4 bytes por elemento)

single Número en punto flotante de precisió simple, con valores aproximadamente en el rango de -10^{38} a 10^{38} (4 bytes por elemento)

char Carácteres (2 byte por elemento)

logical Los valores son 0 ó 1 (1 byte por elemento)

MANEJO BÁSICO DE IMÁGENES CON MATLAB

Tipos de Imágenes:

Imágenes de intensidad: Una matriz cuyos valores han sido escalados para representar intensidad. Pueden ser uint8 ó uint16. Si son double, los valores están escalados entre [0, 1]

Imágenes binarias: Imágenes que solamente tienen valor 0 ó 1. Se representan en Matlab a partir de arrays lógicos. Para convertir en Matlab un array de 0's y 1's en array lógico:

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

Para comprobar si un array es lógico:

```
>>isLogical(A)
```

devuelve un 1 si es lógica y un 0 si no lo es.

Imágenes indexadas.

Imágenes RGB: Se verán más adelante

MANEJO BÁSICO DE IMÁGENES CON MATLAB

Convirtiendo tipos y clases de imágenes:

Comando:	Convirte a:	Tipo válido de entrada:
im2uint8	uint8	logical, uint8, uint16 y double
im2uint16	uint16	logical, uint8, uint16 y double
mat2gray	double ([0,1])	double
im2double	double	logical, uint8, uint16 y double
im2bw	logical	uint8, uint16 y double

```
>> f = [0 0.5; 0.75 1.5]
f =
         0    0.5000
    0.7500    1.5000
>> g = im2uint8(f)
g =
    0  128
  191  255
```

```
>> g = [0 0.3; 0.7 0.9]
g =
      0   0.3000
   0.7000   0.9000
>> gb = im2bw(g, 0.6)
gb =
   0   0
   1   1
```