

# Tema 2: Uso de los comandos básicos

---

## 1. Números y operaciones básicas

El programa trabaja con diferentes tipos de números y expresiones numéricas: números enteros, racionales, reales y complejos.

### Operaciones básicas

Comenzaremos realizando con el programa las operaciones más básicas entre números. Trabajamos en la ventana de Command window y para ello se utilizan las convenciones matemáticas estándar.

Por ejemplo, para sumar (figura 6):

```
>> 2+2
```

```
ans 4
```

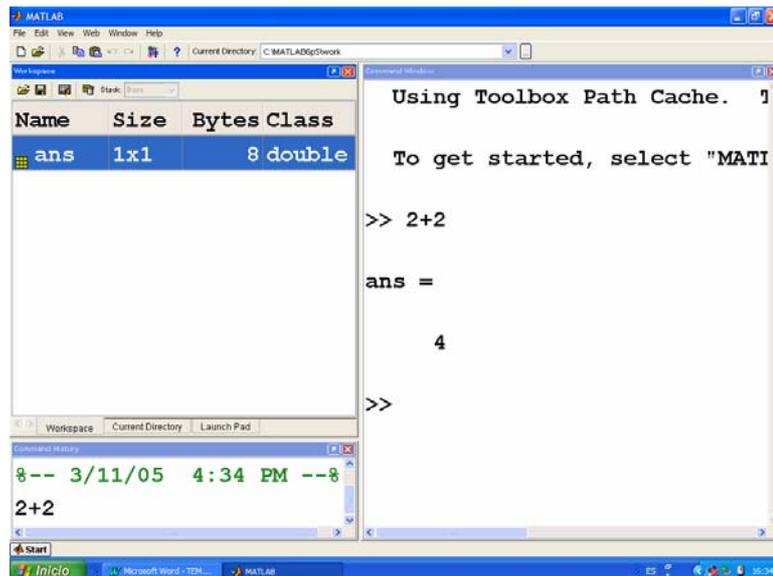


Figura 6

Nota: La respuesta de Matlab se realiza a través de una variable que va cambiando de valor y que guarda el último resultado ejecutado. Verificarlo en el workspace.

En general las operaciones básicas a realizar se hacen según la siguiente tabla y con la jerarquía habitual entre ellas:

$a+b$	Suma
$a-b$	Resta
$a*b$	Producto
$a/b$	División
$a^b$	Potencia

Nota: Por defecto Matlab trabaja con unos dígitos de aproximación para números decimales. Esto puede cambiarse.

### Estableciendo la precisión de los cálculos

MATLAB trabaja siempre en doble precisión, es decir, guardando cada dato en 8 bytes, con 15 cifras decimales exactas. El formato con el que los datos aparecen en pantalla es variable. Existen varios modos de trabajo, se indican a continuación los más significativos:

<b>format short</b>	punto fijo con 4 decimales
<b>format long</b>	punto fijo con 14 decimales
<b>format short e</b>	4 decimales y forma exponencial.
<b>format long e</b>	15 decimales y forma exponencial.
<b>format rat</b>	formato racional.

El modo de trabajo por defecto en MATLAB es `format short`.

Ejemplos:

```
>>3^100
5.1538e+047
>>5+pi
8.1416
```

```

>>format short e
>>3^100
    5.1538e+047
>>5+pi
    8.1416 e+000
>>format long
>>3^100
    5.153775207320113e+047
>>5+pi
    8.14159265358979
>>format long e
>>3^100
    5.153775207320113e+047
>>5+pi
    8.141592653589793e+000
>> format rat
>> 5+pi
    920/113

```

Tecleando format se vuelve al formato por defecto, es decir, format short.

### Números complejos

Matlab integra perfectamente el trabajo con números complejos. La letra minúscula  $i$  ó  $j$  representa el número imaginario  $\sqrt{-1}$  (la unidad imaginaria). De esta manera los números complejos se representan mediante expresiones del tipo  $a+bi$  ó  $a+bj$ .

Existen funciones específicas que actúan sobre los números complejos:

- real(z):** obtiene la parte real de z.
- imag(z):** obtiene la parte imaginaria de z.
- conj(z):** obtiene el complejo conjugado de z.
- abs(z):** obtiene el módulo de z.
- angle(z):** obtiene el argumento de z.

## 2. Variables

Las variables en Matlab tienen forma matricial: fila y columna. Así, si trabajamos con una variable unidimensional se crea con la forma de una fila y una columna. Para hacerlo sólo debemos teclear el nombre elegido e igualarlo a su valor:

```
>> a=8
```

### Alterar el valor de una variable:

Matlab guarda el valor de la variable ejecutada en último lugar, es decir si volvemos a ejecutar un valor para “a” éste será el que mantiene.

Podemos alterar el valor de una variable desde el workspace. Para ello pincharemos en ella en la ventana del workspace y cambiaremos su valor desde el editor (figura 7):

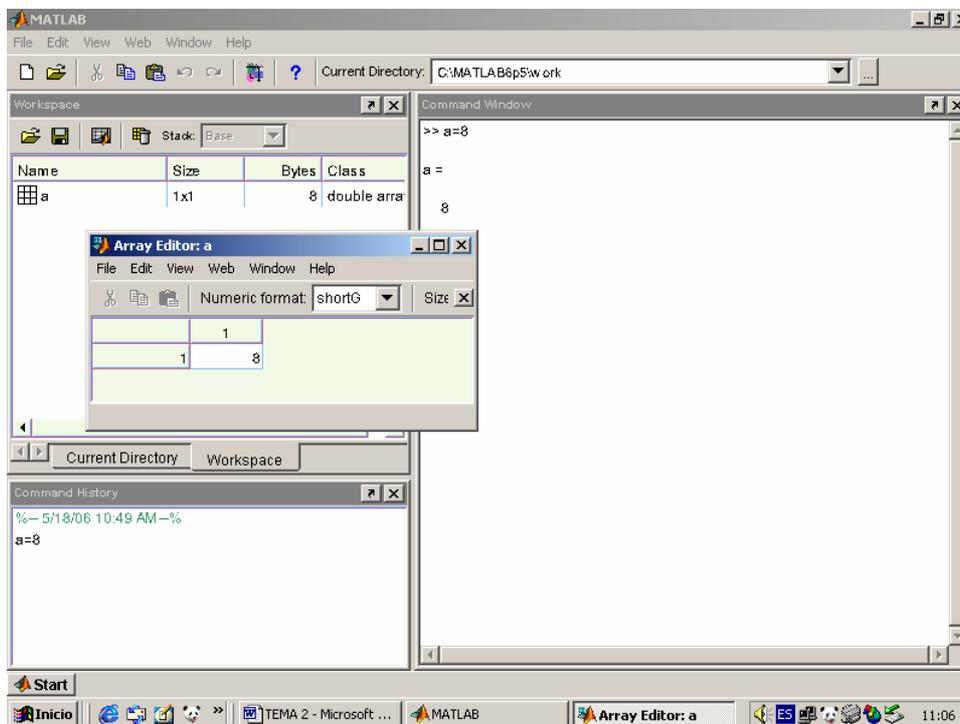


Figura 7

### Guardar variables y recuperarlas

Normalmente es de gran interés guardar los valores de las variables con las que se ha trabajado en una sesión. Bien porque debe interrumpirse la misma y quiere recuperarse más adelante, o bien para utilizarse en nuevos trabajos relacionados con el que se ha realizado.

La forma más básica de guardar las variables es a través del menú File o del botón de guardar situado en la ventana del Workspace (figura8):

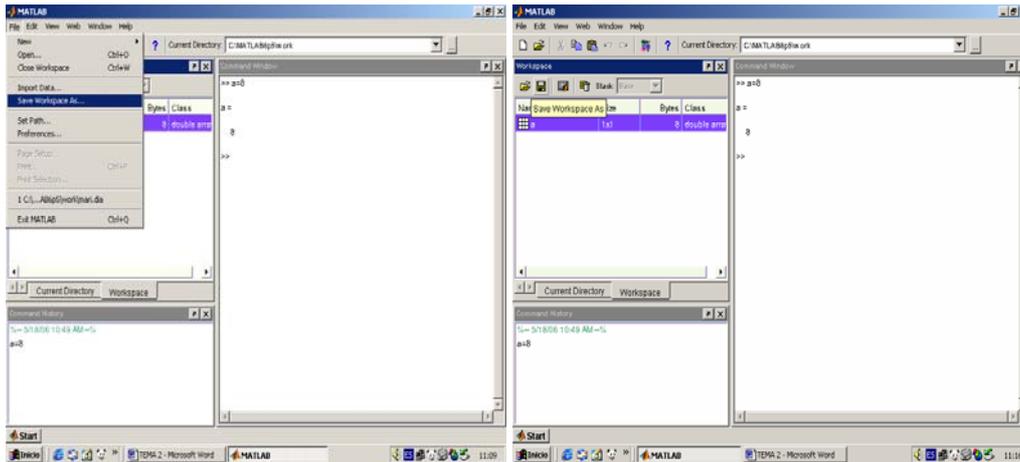


Figura 8

O a través de del menú desplegado al pinchar con el botón derecho del ratón sobre la variable situada en el Workspace (figura 9). Esta opción nos permite guardar sólo las variables seleccionadas. Destacar que esto también ofrece la posibilidad de eliminar la variable. Algo que puede hacerse de forma global desde el menú Edit, submenú Clear Workspace.

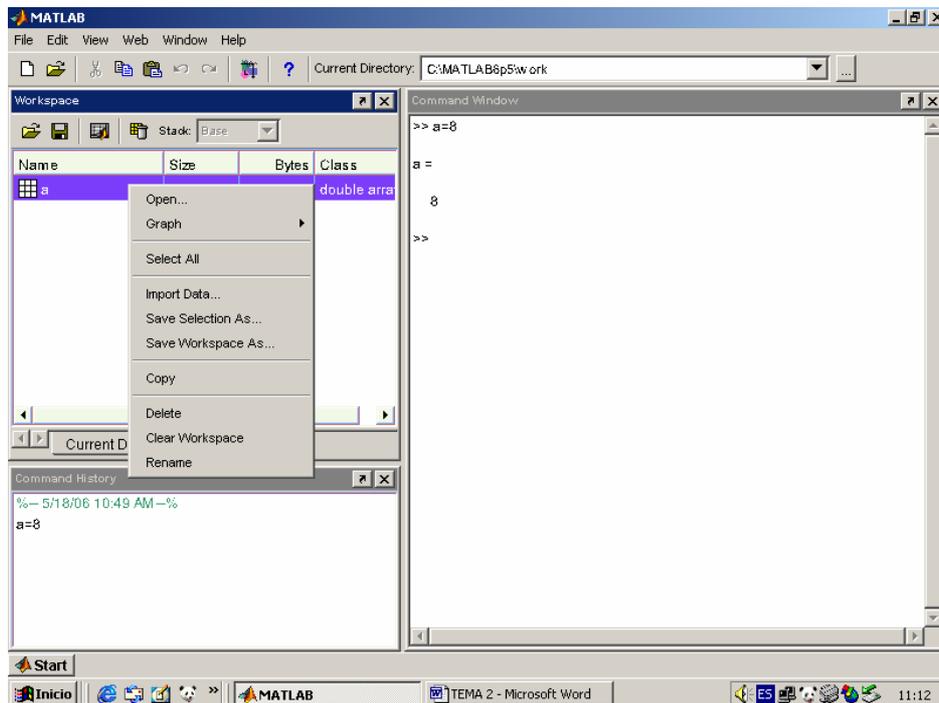


Figura 9

Realizar cualquiera de estos procesos nos permite crear un fichero de extensión .mat con el nombre deseado cuyo contenido serán las variables que se

quieren guardar. Al hacerlo, si lo guardamos en el directorio actual aparecerá dicho fichero en la lista de los que tenemos a nuestra disposición.

Si en una nueva sesión queremos recuperar dichas variables sólo tenemos que cargar el fichero .mat, bien pinchando dos veces sobre su nombre en la ventana del directorio actual o bien a través del menú File, submenú Open.

### 3. Guardar sesiones de trabajo

Existen varios caminos para guardar lo realizado en una sesión de trabajo o en parte de ella:

- Copiar la parte que nos interese del Command Window y abrir un fichero .m. (figura 10).

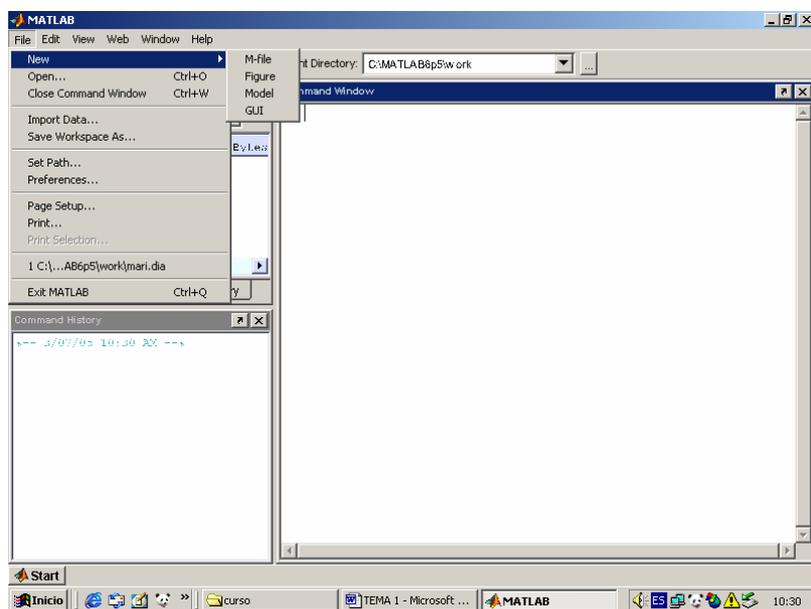


Figura 10

Aparece entonces una nueva ventana (figura 11):

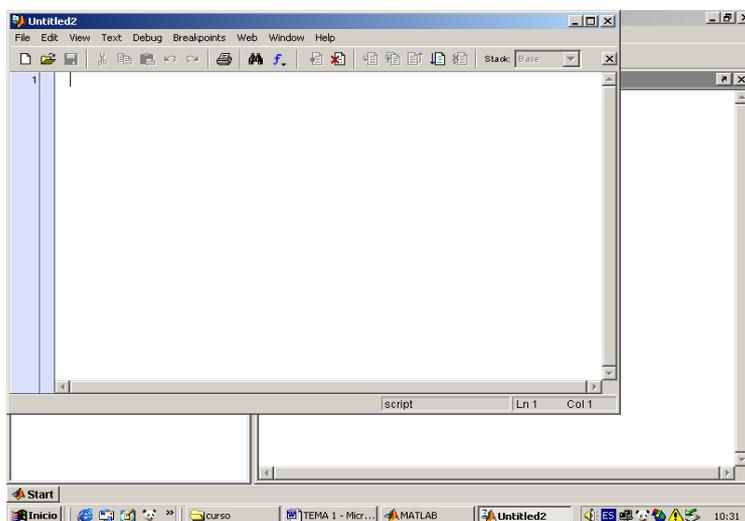


Figura 11

En ella podemos pegar lo seleccionado y modificar cuantas cosas queramos. Este fichero se guarda y se puede acceder a él cuando se desee. Debemos tener en cuenta que según lo realizado se trata de un fichero de texto, no es ejecutable.

- Con los comandos `diary fichero.dia`  
`diary off`

Se trata de escribir al comienzo de la parte que deseamos guardar `diary` seguido del (nombre del fichero donde guardaremos el trabajo).`dia`. Al finalizar terminaremos escribiendo `diary off`. Veremos que se ha creado un fichero que podemos visualizar escribiendo `type fichero`, lo que nos recupera en la pantalla su contenido, o pinchando dos veces sobre él en el `current directory`, lo que nos abre una pantalla con su contenido donde podemos modificar, escribir, etc.

#### 4. Variables carácter

Una cadena de caracteres determinan una variable carácter. Su sintaxis es:  
`a='cadena de caracteres'`

Como vemos es necesario incluir los caracteres entre comillas.

Se destacan algunos comandos importantes que manejan este tipo de variables y que se irán usando en temas sucesivos:

**`str2mat(x1,x2,...)`**: Forma una matriz cuyas filas son las cadenas de caracteres `x1`, `x2`,..., respectivamente.

**`str2num(x1,x2,...)`**: Convierte la cadena de caracteres en su valor numérico exacto utilizado por Matlab.

**`num2str(número)`**: Convierte el número exacto en su cadena de caracteres equivalente con la precisión fijada

**`eval(expresión)`**: Ejecuta la expresión.

**`disp('cadena')`**: Muestra la cadena tal y como se ha escrito y continua el proceso.

**`input('cadena')`**: Muestra la cadena en pantalla y espera que se presión de una tecla para continuar.

Veamos algunos ejemplos:

```
>> num2str(pi)
```

```
ans =
```

```
3.142
```

Se ha convertido el número exacto pi en cadena de caracteres.

```
>> str2num('15/14')
```

```
ans =
```

```
1.0714
```

Se ha convertido una cadena a su valor exacto con la precisión por defecto.

```
>> str2mat('la primera','la dos','telemadrid','antena 3','telecinco')
```

```
ans =
```

```
    la primera
```

```
    la dos
```

```
    telemadrid
```

```
    antena 3
```

```
    telecinco
```

Se ha formado la matriz de texto cuyas filas son las cadenas introducidas como los argumentos de str2mat.

```
>> disp('hola')
```

```
ans =
```

```
hola
```

Se ha obtenido en pantalla la palabra escrita como argumento del comando.

## Práctica 2: Uso de los comandos básicos

1. Calcular  $3+5$  y  $7/8$  en una misma orden.
2. Realiza la operación  $174/13$ .
3. Ofrece la operación anterior con 14 cifras decimales, en forma de número racional y en forma exponencial.
4. Volver al formato short.
5. Definir las siguientes variables:  $a=2 \times 8$ ,  $b=45/6$ ,  $c=a+b$ .
6. Guardar las variables a y b en un fichero.
7. Borrar el valor de todas las variables creadas.
8. Recuperar a y b.
9. Utilizando el comando help, acceder a la ayuda correspondiente a las funciones matemáticas, extrayendo la información necesaria para realizar la siguiente operación:  $c = (\log_2 7)^4 + \sqrt{|b|} - \tan(\pi / 3) + e^a$
10. Asignar a la variable d el valor 34 y calcular el coseno de d al cubo. Hacerlo de forma que se ejecuten las dos operaciones pero sólo se muestre el último resultado.
11. Guardar todas las variables.
12. Guardar la sesión en un fichero .m.
13. El trabajo a realizar en los ejercicios siguientes debe guardarse en un archivo .dia.
14. Introducir los números complejos  $z=2-3i$ ,  $u=-1+2i$ 
  - a. Calcular parte real e imaginaria de z
  - b. Determinar argumento y módulo de u
  - c. Calcular  $z+u$  y  $z \times u$
15. Introducir los complejos  $z=\sqrt{3} + i$ ,  $w=\frac{1}{2} + \frac{\sqrt{3}}{2}i$ . Calcular  $z \times w$  y  $z/w$ .