

BLOQUE TEMÁTICO II

FUNDAMENTOS DE PROGRAMACIÓN Y APLICACIONES EN INGENIERÍA

TEMA 4. Fundamentos de programación

TEMA 5. Programación básica en MATLAB[®] / Octave

TEMA 6. Programación de macros en Excel empleando Visual Basic

TEMA 5. PROGRAMACIÓN BÁSICA EN MATLAB[®]/OCTAVE

5.1. Introducción a Matlab y Octave

5.2. Entrada y salida con formato

5.3. Programas: script y funciones

5.4. Estructuras alternativas o condicionales

5.5. Estructuras repetitivas o bucles

5.6. Aplicación de MATLAB[®] / Octave a casos prácticos de ingeniería

PROGRAMACIÓN EMPLEANDO PSEUDOCÓDIGO Y ORDINOGRAMAS

ALTERNATIVAS

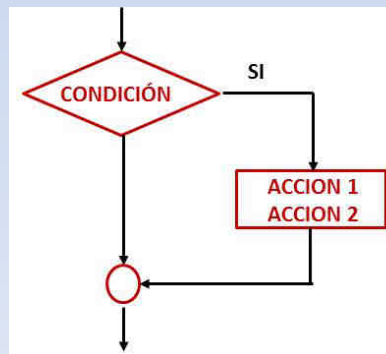


Controlan la ejecución de uno o varios bloques de instrucciones dependiendo del cumplimiento o no de alguna condición o valor final de una expresión (simple, doble y múltiple)

Simple



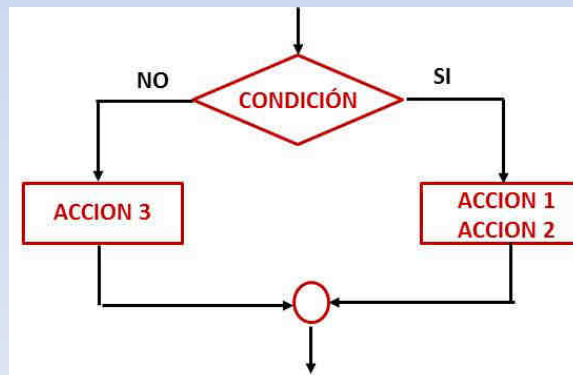
*Si (expresión) entonces
Accion 1
Acción 2
Fin si*



Doble



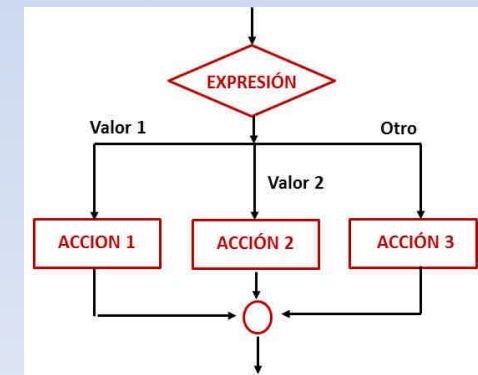
*Si (expresión) entonces
Accion 1
Acción 2
Si no
Accion 3
Fin si*



Múltiple



*Según sea (expresión)
Valor 1
Instruccion 1
Valor 2
Instruccion 2
Otros
Instruccion 3
Fin según sea*



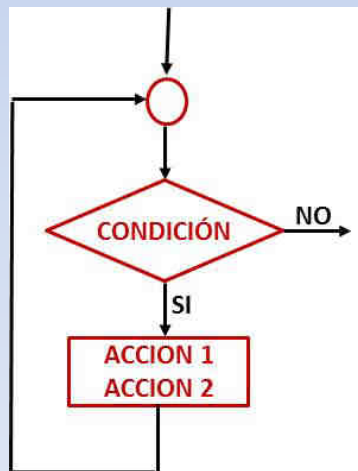
PROGRAMACIÓN EMPLEANDO PSEUDOCÓDIGO Y ORDINOGRAMAS

REPETITIVAS

Controlan la repetición de un conjunto de instrucciones mediante la evaluación de una condición, la cual se realiza en cada nueva repetición o por medio de un contador (mientras, repetir y para)

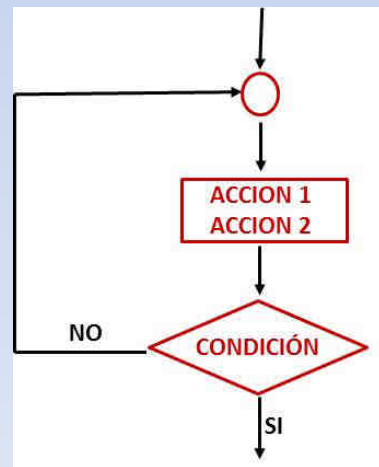
Mientras (while)

*Mientras (condición)
Acción 1
Acción 2
Fin_mientras*



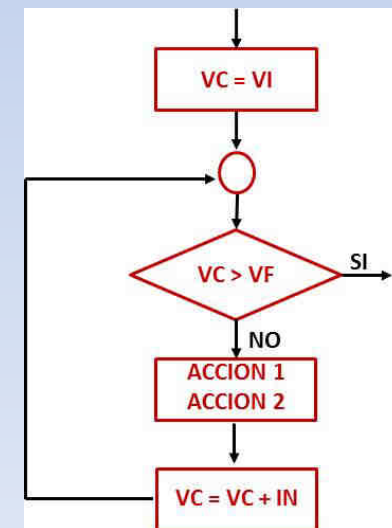
Repetir (do until)

*Repetir
Acción 1
Acción 2
Hasta (condición)
Fin_repetir*



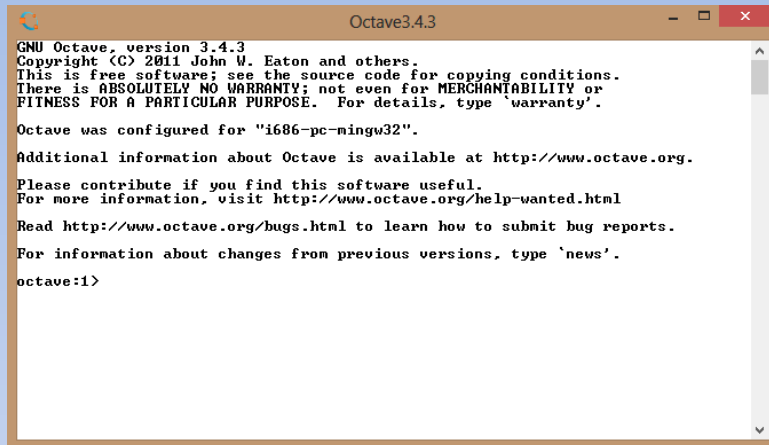
Para (for)

*Para VC = VI :IN:VF hacer
Acción 1
Acción 2
Fin_para*



MATLAB/OCTAVE COMO CALCULADORA

<http://gnu.org/software/octave/>



```
GNU Octave, version 3.4.3
Copyright (C) 2011 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "i686-pc-mingw32".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

octave:1>
```

VARIABLES

REGLAS DE ASIGNACIÓN VARIABLES PREDEFINIDAS

ans	Variable que almacena el último resultado.
pi	Valor $\pi = 3.1415\dots$
eps	Menor número que, sumado a 1, da un resultado mayor que 1.
inf	Infinito, p.e., $1/0$.
NaN o nan	Resultado no numérico (<i>Not a number</i>), p.e., $0/0$.
i y j	$\sqrt{-1}$.
e	Base de los logaritmos naturales (neperianos).

CÁLCULOS SIMPLES

Operador	Símbolo	Ejemplo
suma, $a + b$	+	$3 + 22$
resta, $a - b$	-	$90 - 54$
multiplicación, $a * b$	*	$3.14 * 0.85$
división, $a \div b$	/ o \	$56/8$
exponenciación, a^b	^	2^8
cambio de signo, $-a$	-	$-(4*8)$

OPERADORES DE COMPARACIÓN

Operador	Símbolo	Ejemplo
Menor que	<	$2 < 3$
Mayor que	>	$3 > 2$
Menor o igual que	<=	$3 <= 4$
Mayor o igual que	>=	$3 >= 2$
Igual a	==	$\text{Inf} == \text{Inf} + 1$
Distinto de	~=	$3 \sim 4$

~ alt+126

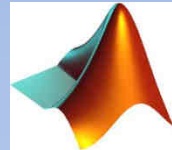
MANEJO DE VECTORES Y MATRICES

GRÁFICAS EN 2D Y 3D

AJUSTES DE CURVAS

PROGRAMACIÓN EN MATLAB/OCTAVE

MATLAB

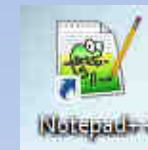


OCTAVE



+ EXPLORADOR DE WINDOWS

+ NOTEPAD ++

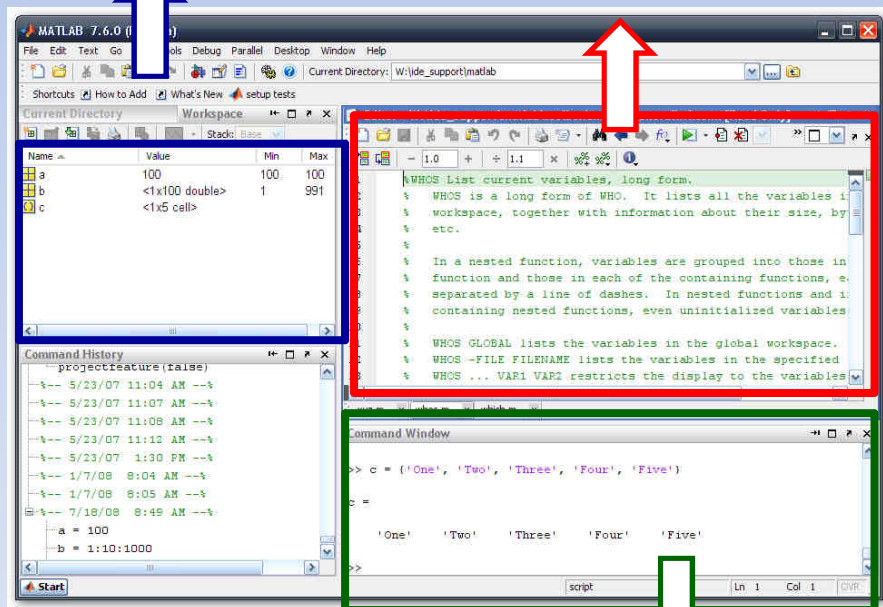


**FICHEROS DE MATLAB
EN LA CARPETA ACTIVA**

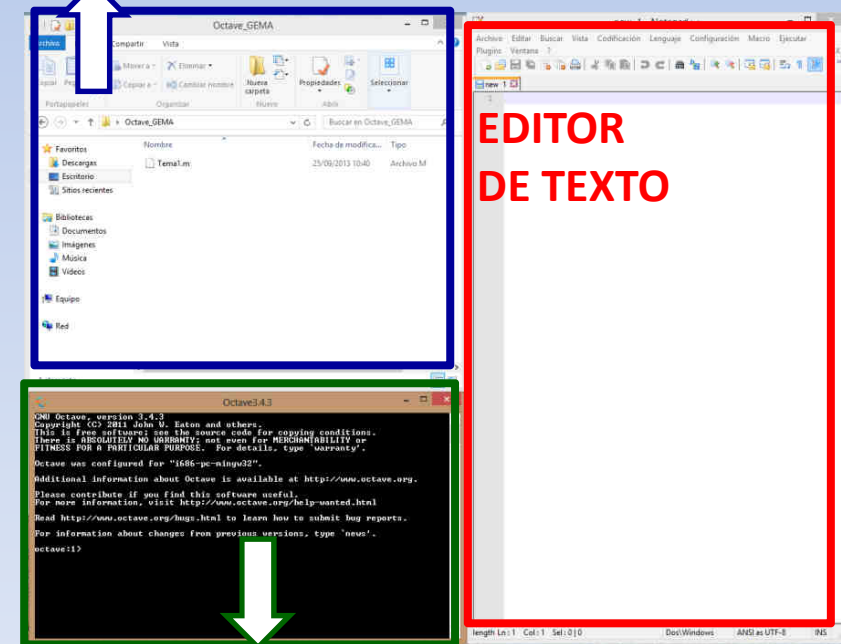
EDITOR DE TEXTO

FICHEROS DE OCTAVE

**EDITOR
DE TEXTO**



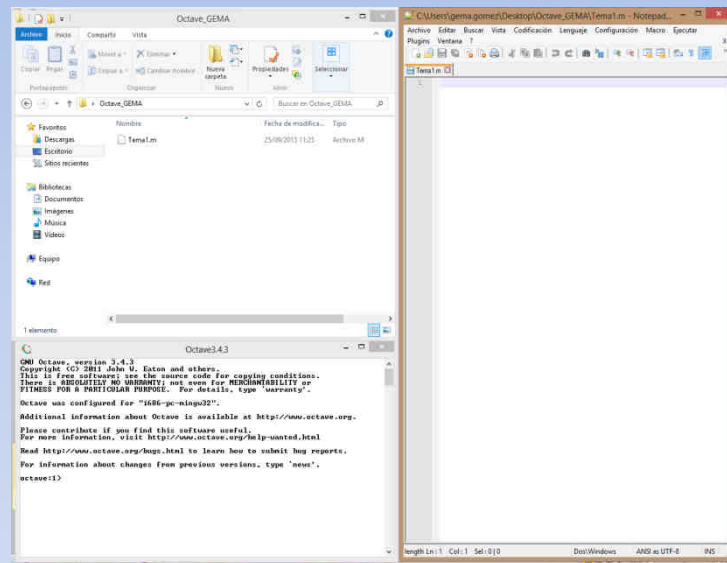
VENTANA DE COMANDOS



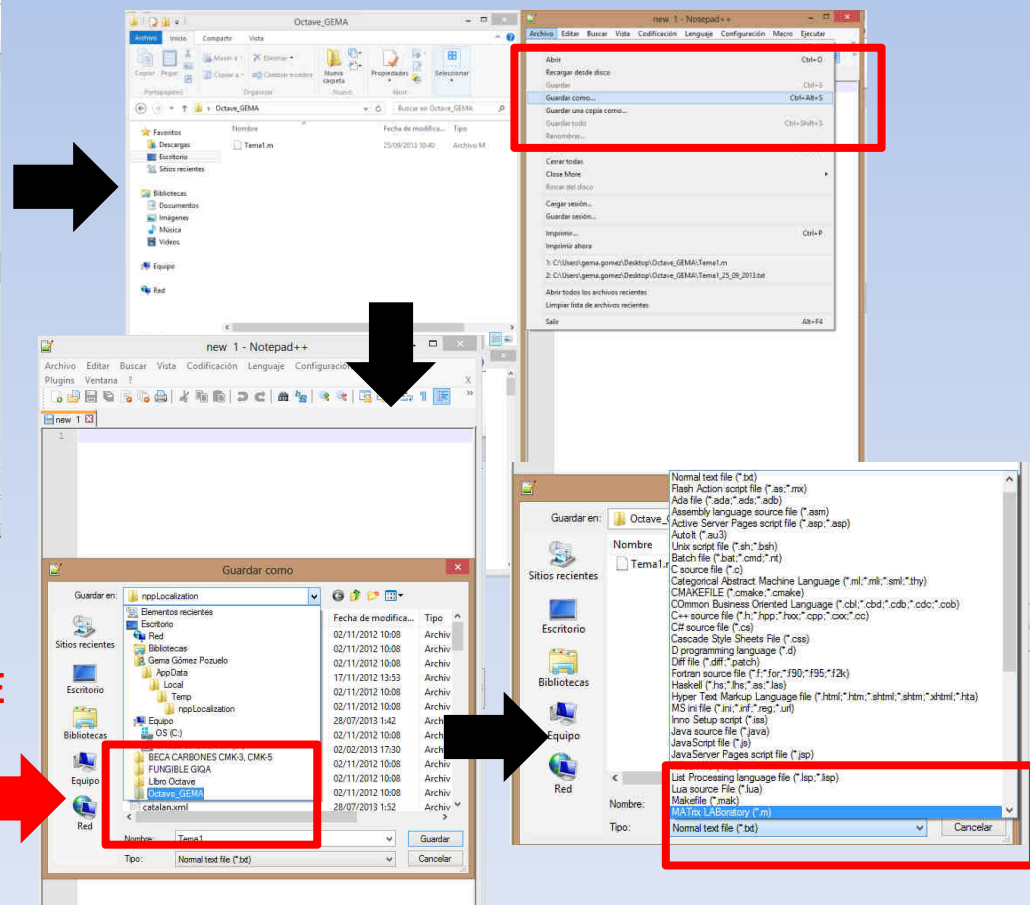
VENTANA DE COMANDOS

¿CÓMO TRABAJAR CON OCTAVE y NOTEPAD++?

1. ABRIR OCTAVE, NOTEPAD ++ Y EL EXPLORADOR DE WINDOWS



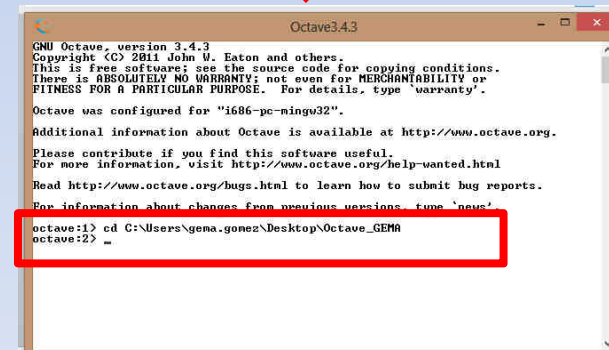
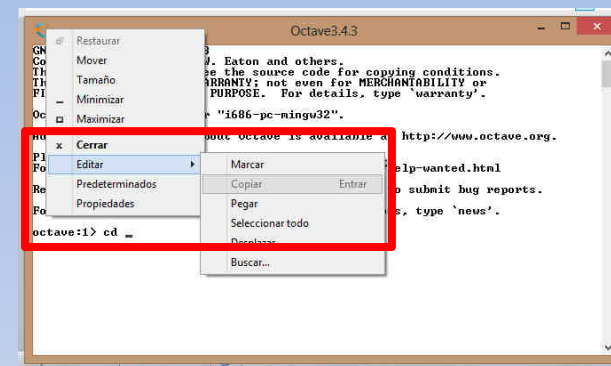
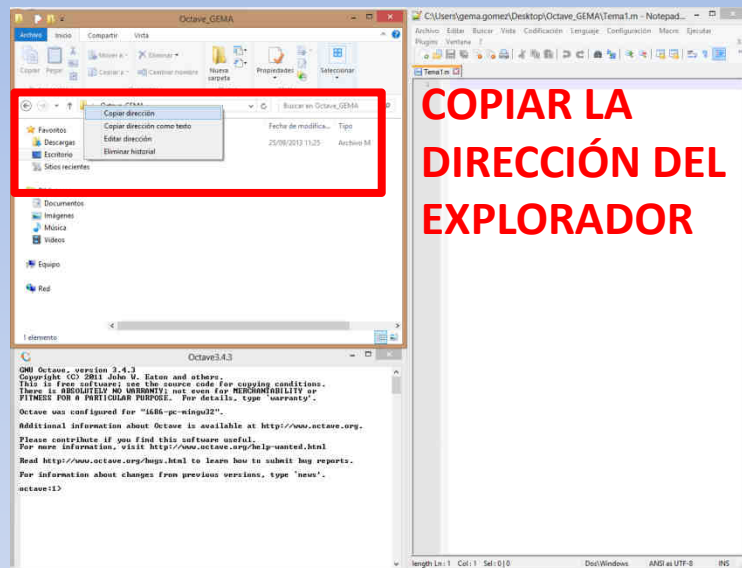
2. GUARDAR EL ARCHIVO EN NOTEPAD ++ CON EXTENSION .m EN LA CARPETA ELEGIDA DEL EXPLORADOR



**EL NOMBRE DE LA CARPETA NO DEBE
CONTENER ESPACIOS EN BLANCO O
CARACTERES ESPECIALES**

¿CÓMO TRABAJAR CON OCTAVE y NOTEPAD++?

3. CAMBIAR EL DIRECTORIO EN LA VENTANA DE COMANDOS DE OCTAVE



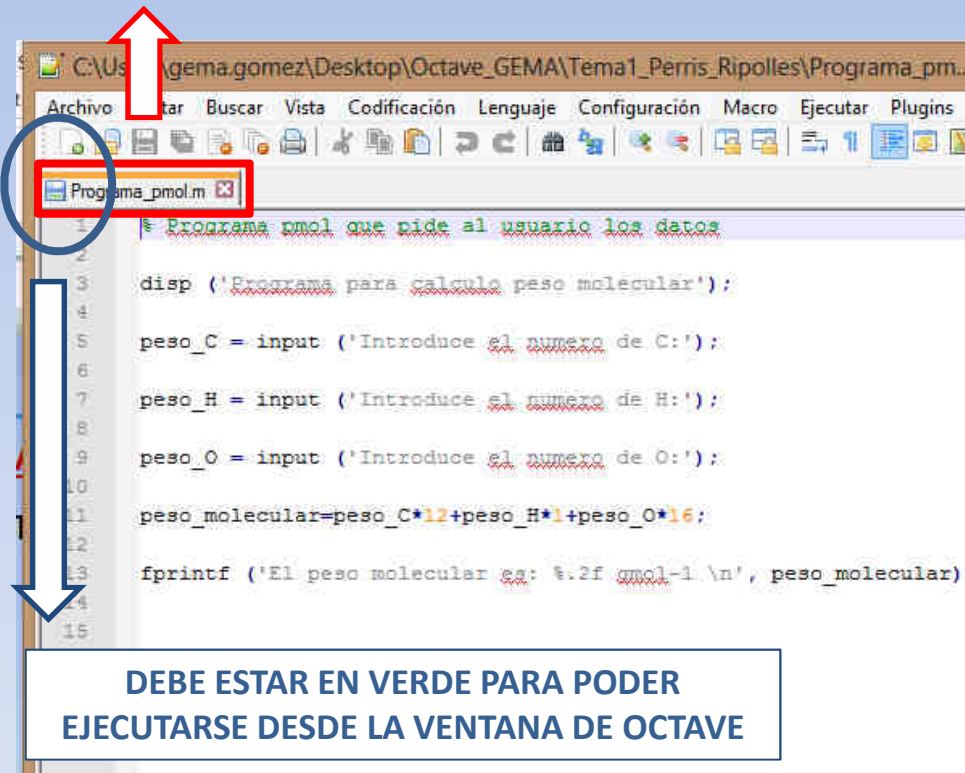
ANTES DE LA DIRECCIÓN ESCRIBIR LA PALABRA `cd` SEGUIDO DE LA DIRECCIÓN
(la dirección no debe contener espacios ni caracteres diferentes de letras o “_”)

ESTOS PASOS DEBEN REPETIRSE CADA VEZ QUE VAYAMOS A USAR OCTAVE

¿CÓMO TRABAJAR CON OCTAVE y NOTEPAD++?

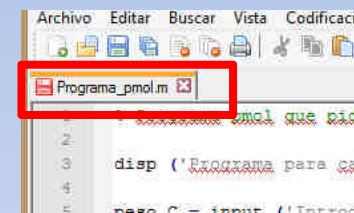
4. ESCRIBIR LOS COMANDOS EN NOTEPAD ++

GUARDAR CON UN NOMBRE SIN ESPACIOS

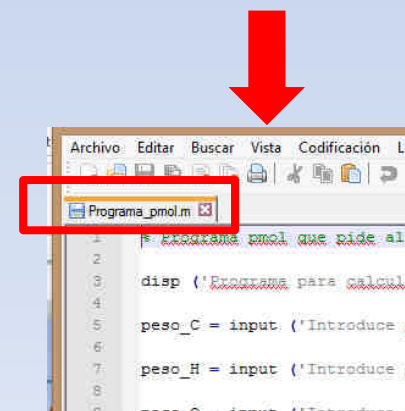


DEBE ESTAR EN VERDE PARA PODER EJECUTARSE DESDE LA VENTANA DE OCTAVE

SI MODIFICAMOS ALGO EN EL ARCHIVO APARECE ICONO EN ROJO

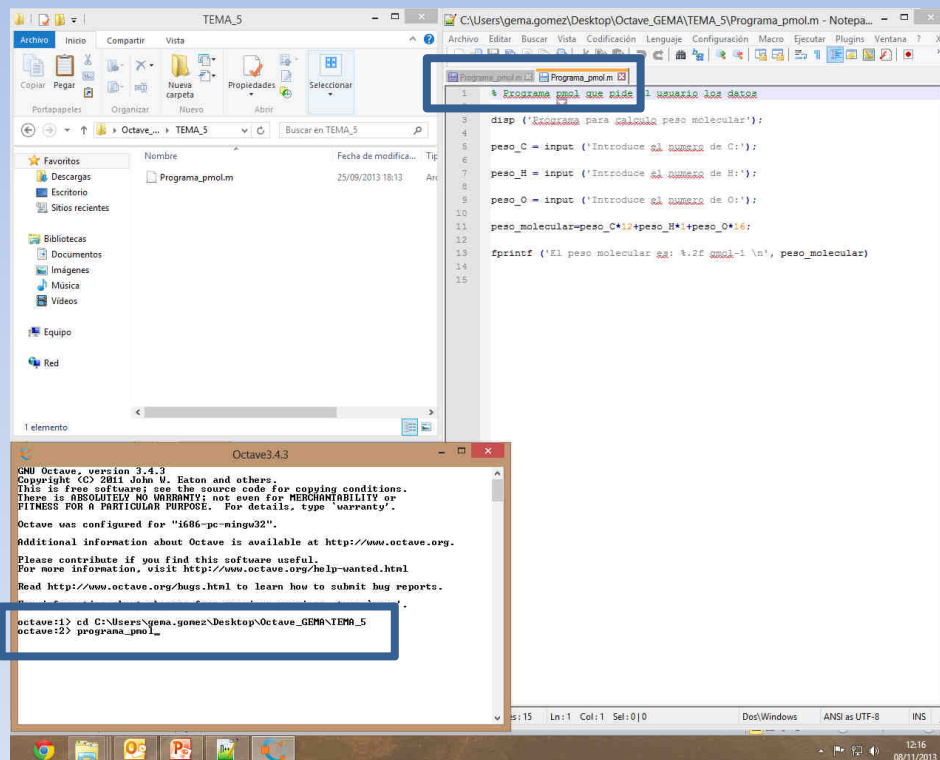


ANTES DE EJECUTAR EN OCTAVE, DARLE A GUARDAR

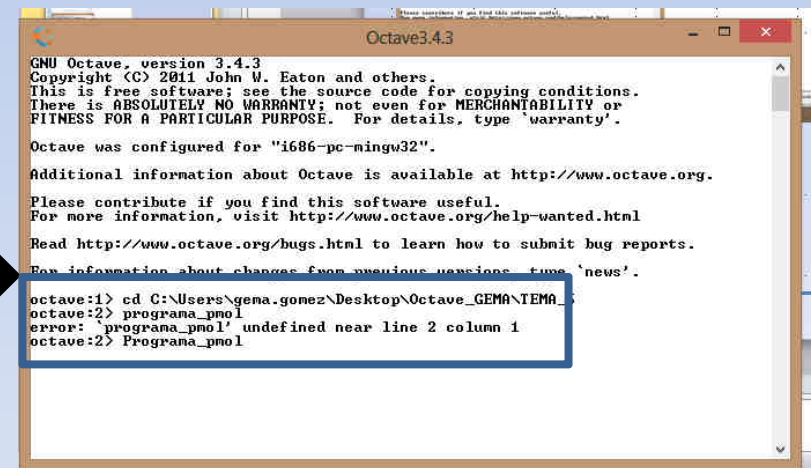


¿CÓMO TRABAJAR CON OCTAVE y NOTEPAD++?

5. EJECUTAR EL PROGRAMA EN VENTANA DE OCTAVE ESCRIBIENDO EL NOMBRE DEL ARCHIVO DE NOTEPAD ++

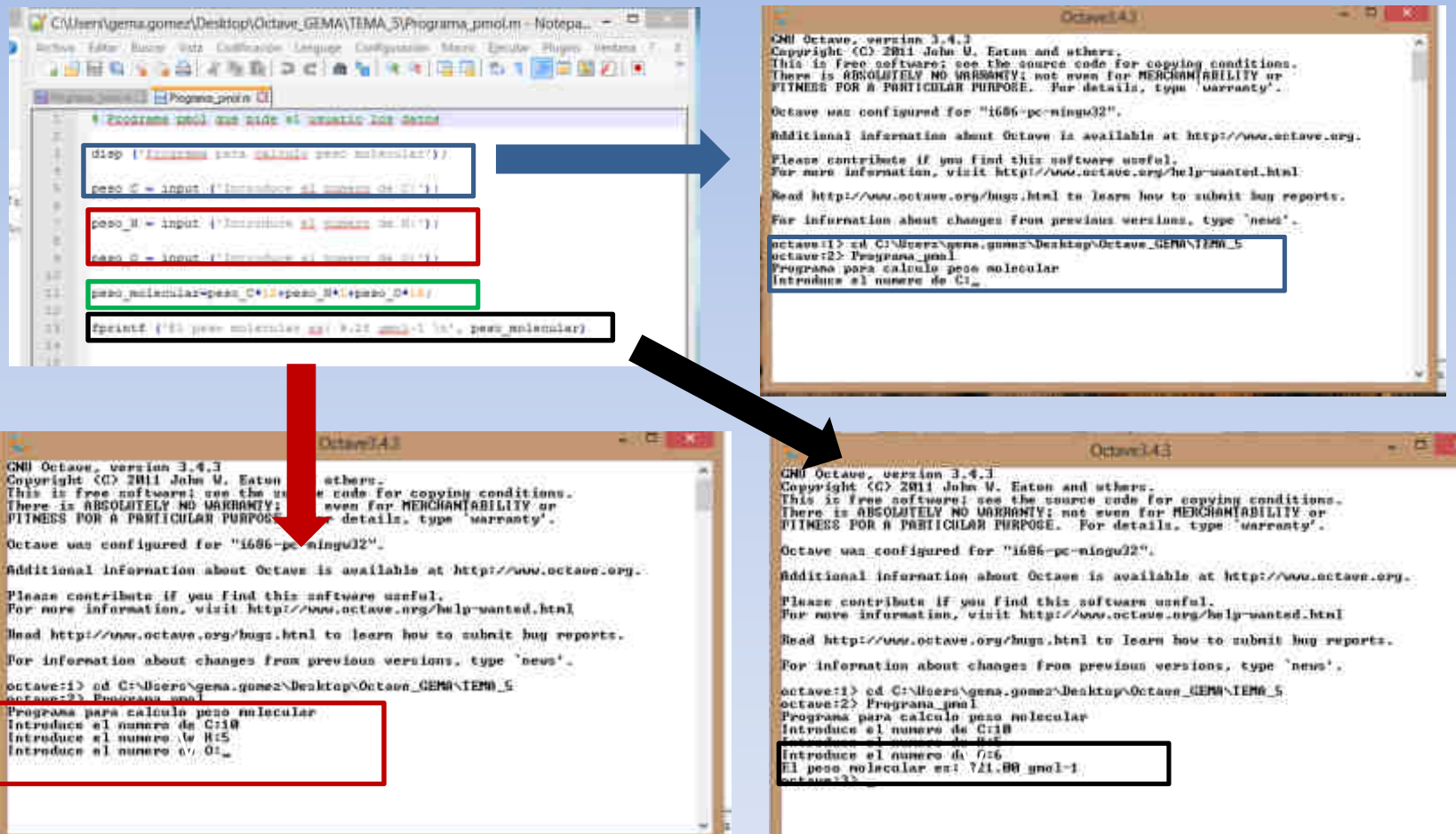


Escribir nombre programa en octave,
diferencia entre mayúsculas y
minúsculas



¿CÓMO TRABAJAR CON OCTAVE y NOTEPAD++?

6. COMPROBAR QUE EL PROGRAMA REALIZADO NO CONTIENE ERRORES



RESOLUCIÓN DE PROBLEMAS EN INGENIERÍA Y CIENCIAS

- 1. PLANTEAR EL PROBLEMA.**
- 2. DESCRIBIR LOS VALORES DE ENTRADA (CONOCIDOS) Y LAS SALIDAS (INCOGNITAS) QUE SE REQUIEREN.**
Tener en cuenta las unidades e identificar las constantes que son necesarias para la resolución del problema.
- 3. DESARROLLAR EL ALGORITMO (pseudocódigo u ordinograma).**
- 4. RESOLVER EL PROBLEMA CREANDO EL CÓDIGO EN MATLAB / OCTAVE.**
- 5. PROBAR QUE NO EXISTEN ERRORES EN EL CÓDIGO Y QUE LA SOLUCION ES CORRECTA (LAS GRÁFICAS SON ÚTILES PARA VERIFICAR QUE LOS RESULTADOS SON COHERENTES).**

TEMA 5. PROGRAMACIÓN BÁSICA EN MATLAB[®]/OCTAVE

5.1. Introducción a Matlab y Octave

5.2. Entrada y salida con formato

5.3. Programas: script y funciones

5.4. Estructuras alternativas o condicionales

5.5. Estructuras repetitivas o bucles

5.6. Aplicación de MATLAB[®] / Octave a casos prácticos de ingeniería

ENTRADA DEFINIDA POR EL USUARIO

INPUT



Proporciona una cadena de texto en la ventana de comandos para que el usuario proporcione la entrada solicitada.

EN NOTEPAD++

`z = input ('Introduzca un valor')`

```

1  % LA FUNCION INPUT SIRVE PARA PEDIR DATOS AL USUARIO
2
3  % PUEDE SER UN ÚNICO VALOR
4
5  x = input('Introduzca un valor:');
6
7  % UNA MATRIZ
8  % pide al usuario que introduzca una matriz como [1,2,3;4,5,6]
9
10 y = input('Introduzca valores para z entre corchetes:');
11
12 % CARACTERES, PARA ELLO INDICAR QUE INTRODUZCA EL VALOR ENTRE APÓSTROFES
13 % pide al usuario introduzca 'Gema'
14
15 z = input('Introduzca su nombre entre apostrofes:');
16
17 % PARA QUE EL USUARIO NO TENGA QUE PONER APOSTROFES: 's'
18
19 t = input('Introduzca su nombre:', 's')
  
```

INTRODUCIR UN SOLO VALOR

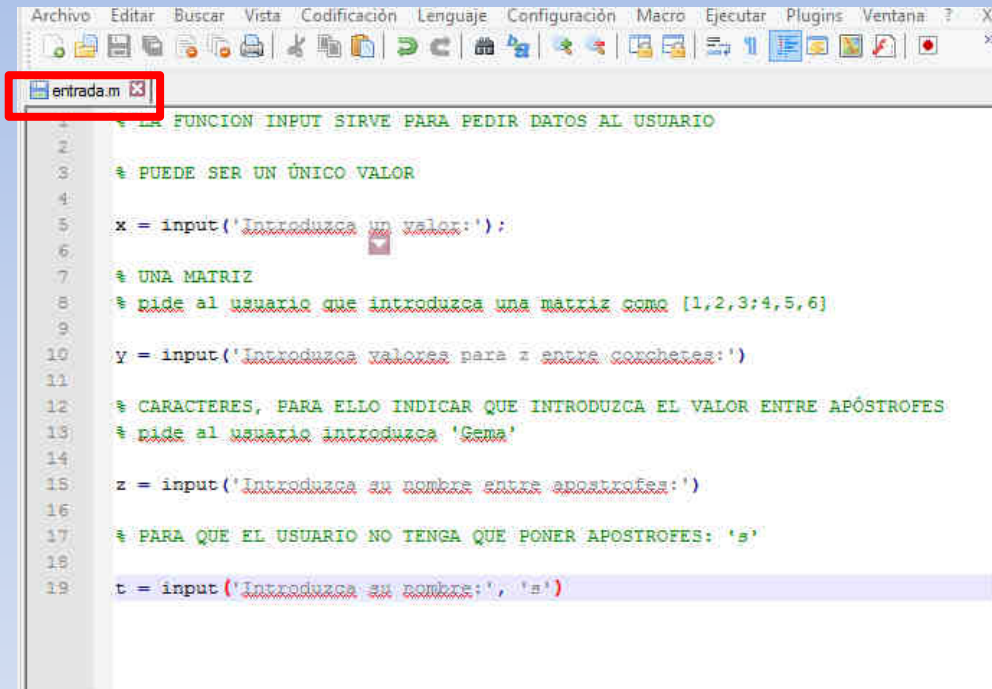
MATRICES

**CADENA DE CARACTERES
(DEBEN APARECER ENTRE COMILLAS o poner 's')**

ENTRADA DEFINIDA POR EL USUARIO

INPUT

EN OCTAVE

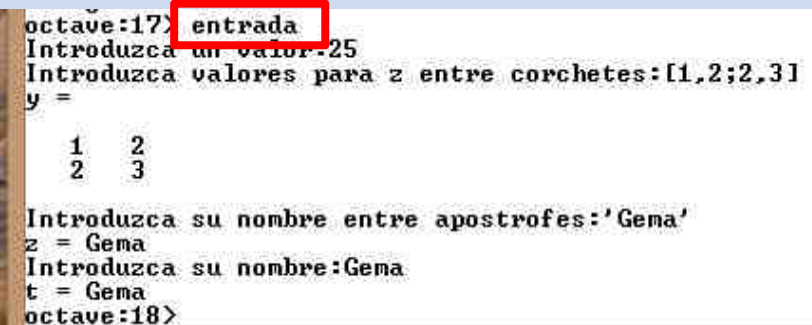


```

1 % LA FUNCION INPUT SIRVE PARA PEDIR DATOS AL USUARIO
2
3 % PUEDE SER UN ÚNICO VALOR
4
5 x = input('Introduzca un valor:');
6
7 % UNA MATRIZ
8 % pide al usuario que introduzca una matriz como [1,2,3;4,5,6]
9
10 y = input('Introduzca valores para z entre corchetes:');
11
12 % CARACTERES, PARA ELLO INDICAR QUE INTRODUZCA EL VALOR ENTRE APÓSTROFES
13 % pide al usuario introduzca 'Gema'
14
15 z = input('Introduzca su nombre entre apostrofes:');
16
17 % PARA QUE EL USUARIO NO TENGA QUE PONER APOSTROFES: 's'
18
19 t = input('Introduzca su nombre:', 's');
  
```

EJECUTAR EL PROGRAMA HECHO EN NOTEPAD++.

LOS COMENTARIOS (%) NO APARECEN EN OCTAVE.



```

octave:17> entrada
Introduzca un valor:25
Introduzca valores para z entre corchetes:[1,2;2,3]
y =

     1     2
     2     3

Introduzca su nombre entre apostrofes:'Gema'
z = Gema
Introduzca su nombre:Gema
t = Gema
octave:18>
  
```

SALIDAS POR PANTALLA

Función DISP



Proporciona una cadena de texto para informar al usuario del uso del programa:

`disp('texto')`

También podemos mostrar el valor de una variable calculada quitando los apostrofes:

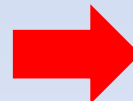
`disp(nombre_variable)`

EN NOTEPAD++

```
1 % INTRODUCIMOS TEXTO
2
3 disp('Este programa calcula el peso molecular')
4
5 % PROPORCIONAMOS EL VALOR DE UNA VARIABLE
6
7 t=5
8
9 disp(t)
10
```

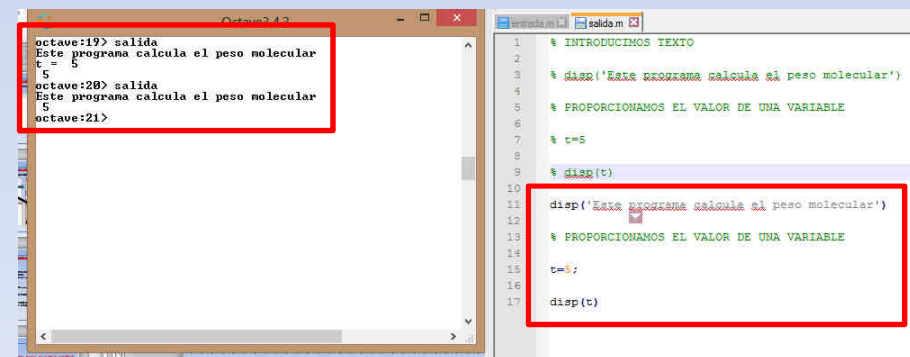


Si no queremos una asignación de variables o constantes o una operación, finalizar la frase con ;



EN OCTAVE

```
error: salida undefined near line 10 column 1
octave:18> salida
Este programa calcula el peso molecular
t = 5
5
octave:19>
```



```
octave:19> salida
Este programa calcula el peso molecular
t = 5
5
octave:20> salida
Este programa calcula el peso molecular
5
octave:21>
```

```
1 % INTRODUCIMOS TEXTO
2
3 disp('Este programa calcula el peso molecular')
4
5 % PROPORCIONAMOS EL VALOR DE UNA VARIABLE
6
7 t=5
8
9 disp(t)
10
11 disp('Este programa calcula el peso molecular')
12
13 % PROPORCIONAMOS EL VALOR DE UNA VARIABLE
14
15 t=5;
16
17 disp(t)
```


EJERCICIO PROPUESTO 1. ESCRIBE UN PROGRAMA LLAMADO operbas.m QUE PIDA AL USUARIO DOS NÚMEROS Y CALCULE CON ELLOS LAS OPERACIONES BASICAS DE SUMA, RESTA, MULTIPLICACIÓN Y DIVISIÓN, MOSTRANDO LOS RESULTADOS POR PANTALLA.

```

1  % programa que pide dos numeros y calcule +, -, * y /
2  % los resultados se muestran por pantalla
3
4  disp ('Programa calculo operaciones basicas')
5
6  numero_1 = input ('Introduzca el primer numero:');
7
8  numero_2 = input ('Introduzca el segundo numero:');
9
10 % poner ; despues del comando input para que no repita el
11 % nombre de la variable despues
12
13 suma = numero_1 + numero_2;
14
15 disp ('La suma es:');
16 disp (suma)
17
18 resta = numero_1 - numero_2;
19
20 disp ('La resta es:');
21 disp (resta)
22
23 mult = numero_1 * numero_2;
24
25 disp ('La multiplicacion es:');
26 disp (mult)
27
28 div = numero_1 / numero_2;
29
30 disp ('La division es:');
31 disp (div)

```

AL PONER ";" NO SE
MUESTRA POR PANTALLA

```

octave-3.2.4.exe:3>
octave-3.2.4.exe:3> E:\propuesto1_T5
Programa calculo operaciones basicas
Introduzca el primer numero:85
Introduzca el segundo numero:25
La suma es:
110
La resta es:
60
La multiplicacion es:
2125
La division es:
3.4000
octave-3.2.4.exe:4>

```

EJERCICIO PROPUESTO 2. ESCRIBE UN PROGRAMA LLAMADO `gasideal.m` QUE UTILICE LA ECUACIÓN DE LOS GASES IDEALES DE MODO QUE PIDA AL USUARIO EL NÚMERO DE MOLES DE UN GAS, LA TEMPERATURA Y EL VOLUMEN A LOS QUE SE ENCUENTRA Y LE PROPORCIONE LA PRESION.

```

1  % programa gases ideales
2
3  disp('Programa para calcular la P de un gas ideal')
4
5  n=input('Introduzca el numero de moles:');
6
7  T=input('Introduzca la temperatura en kelvin:');
8
9  V=input('Introduzca el volumen:');
10
11  R=0.082 %atL/kmol
12
13  P=(n*R*T)/V;
14
15  disp('La presion es:')
16
17  disp(P)
18
19  format long
20
21  disp(P)
22

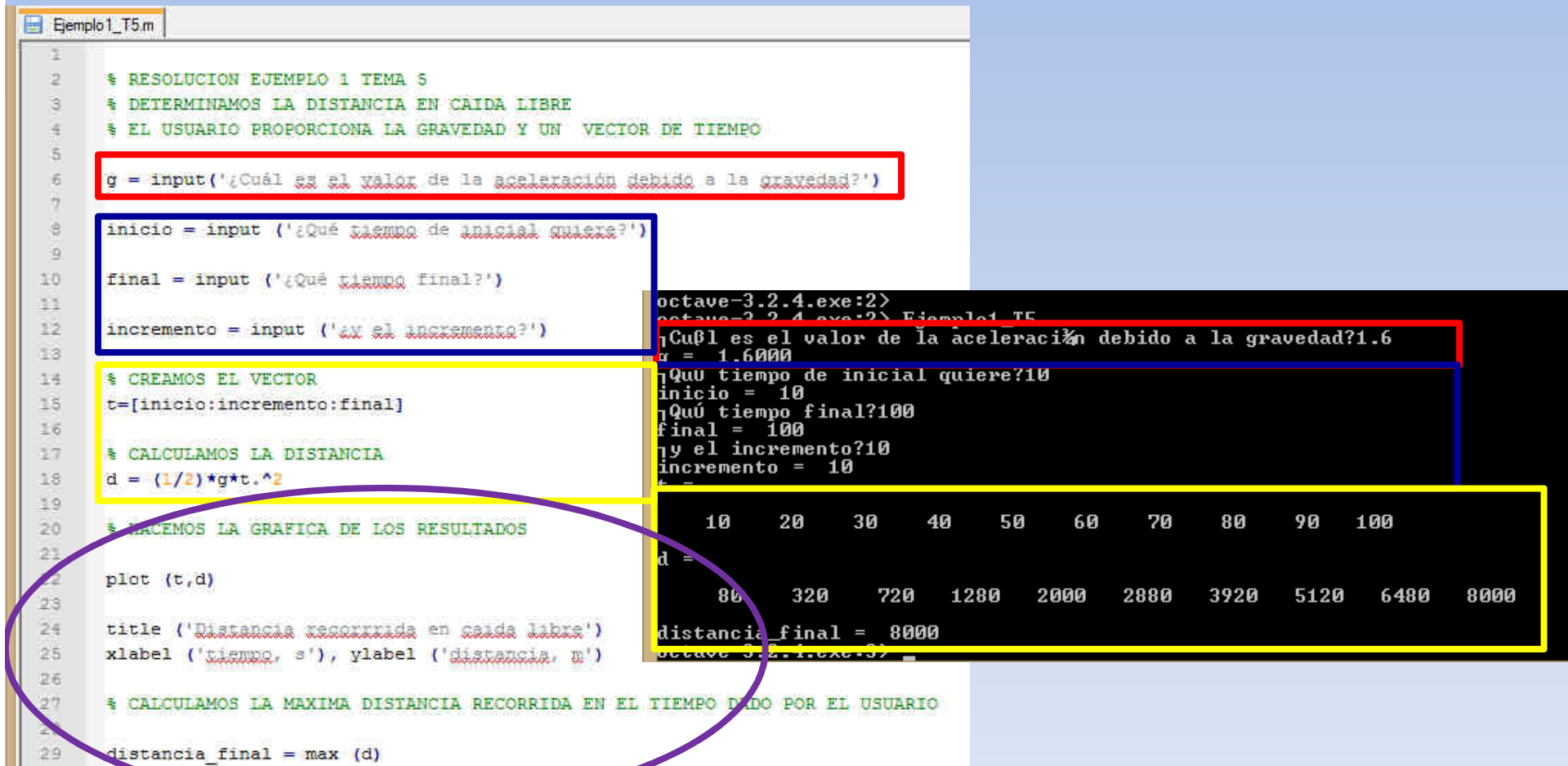
```

```

octave-3.2.4.exe:5>
octave-3.2.4.exe:5> Ejpropuesto2_T5
Programa para calcular la P de un gas ideal
Introduzca el numero de moles:25
Introduzca la temperatura en kelvin:596
Introduzca el volumen:55
R = 0.082000
La presion es:
22.215
22.2145454545455
octave-3.2.4.exe:6>

```

Ejemplo 1. (Input): Analizar el comportamiento de un objeto en caída libre. La ecuación que lo describe es: $d = (1/2) \cdot g \cdot t^2$, donde d es la distancia que recorre el objeto, g es la aceleración debida a la gravedad y t es el tiempo en que el objeto recorre la distancia d . El usuario introduce el valor de g y un vector de valores de tiempo.



```

1  Ejemplo1_T5.m
2  % RESOLUCION EJEMPLO 1 TEMA 5
3  % DETERMINAMOS LA DISTANCIA EN CAIDA LIBRE
4  % EL USUARIO PROPORCIONA LA GRAVEDAD Y UN VECTOR DE TIEMPO
5
6  g = input('¿Cuál es el valor de la aceleración debido a la gravedad?')
7
8  inicio = input('¿Qué tiempo de inicial quiere?')
9
10 final = input('¿Qué tiempo final?')
11
12 incremento = input('¿y el incremento?')
13
14 % CREAMOS EL VECTOR
15 t=[inicio:incremento:final]
16
17 % CALCULAMOS LA DISTANCIA
18 d = (1/2)*g*t.^2
19
20 % HACEMOS LA GRAFICA DE LOS RESULTADOS
21
22 plot (t,d)
23
24 title ('Distancia recorrida en caída libre')
25 xlabel ('tiempo, s'), ylabel ('distancia, m')
26
27 % CALCULAMOS LA MAXIMA DISTANCIA RECORRIDA EN EL TIEMPO DADO POR EL USUARIO
28
29 distancia_final = max (d)
  
```

```

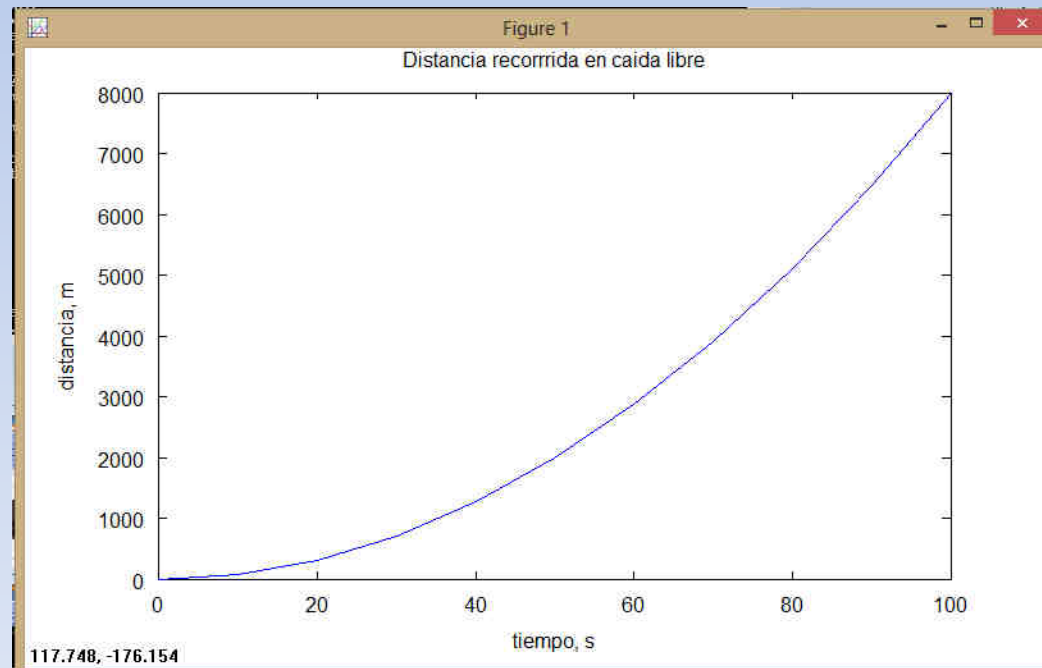
octave-3.2.4.exe:2>
octave-3.2.4.exe:2> Ejemplo1_T5
¿Cuál es el valor de la aceleración debido a la gravedad?1.6
g = 1.6000
¿Qué tiempo de inicial quiere?10
inicio = 10
¿Qué tiempo final?100
final = 100
¿y el incremento?10
incremento = 10
t =
    10    20    30    40    50    60    70    80    90   100
d =
    80    320    720   1280   2000   2880   3920   5120   6480   8000
distancia_final = 8000
octave-3.2.4.exe:3>
  
```

Ejemplo 1b: Analizar el comportamiento de un objeto en caída libre. La ecuación que lo describe es: $d = (1/2) \cdot g \cdot t^2$, donde d es la distancia que recorre el objeto, g es la aceleración debida a la gravedad y t es el tiempo en que el objeto recorre la distancia d .

El usuario introduce el valor de g y un vector de valores de tiempo.

```
plot (t,d)

title ('Distancia recorrida en caída libre')
xlabel ('tiempo, s'), ylabel ('distancia, m')
```



SALIDAS POR PANTALLA

FPRINTF



Proporciona un mayor control de las salidas por pantalla que “disp” ya que permite especificar el formato con el que se van a mostrar los valores. Permite con un solo comando dar tanto caracteres como información numérica.

fprintf (formato, expresiones)

- Formato: contiene el texto (Ejemplo: El valor de la presión es...).
- Expresiones. Especificación del formato, existen varias opciones:

%e: los valores se proporcionan en notación exponencial.

%f: los valores se proporcionan en notación decimal.

%g: los valores se mostrarán en el formato más corto de los anteriores.

%c: para información de tipo carácter.

%s: para tipo cadena de caracteres

- Seguido de las variables que quieran visualizarse separadas por comas.
- Empleando **\n** la acción resultante es un cambio de línea (enter).

FPRINTF

PARA DAR POR PANTALLA LOS RESULTADOS CON EL FORMATO ESTABLECIDO POR DEFECTO Y SIN SALTO DE LÍNEA (SIN \n)

```
ej1.m
1  % EJEMPLO DE USO DE PRINTF
2
3  Re = 1500
4
5  fprintf('El numero de Reynold es:',Re)
6
7  Re = 13000
8
9  fprintf('El numero de Reynold es:',Re)
10
11
```

AL NO PONER “;” SE MUESTRA POR PANTALLA

```
octave-3.2.4.exe:6>
octave-3.2.4.exe:6> ej1
Re = 1500
El numero de Reynold es:Re = 13000
El numero de Reynold es:octave-3.2.4.exe:7>
```

NO HAY SALTO DE LINEA Y “octave 3.2.4>” SE MUESTRA A CONTINUACIÓN

CON SALTO DE LÍNEA (\n)

```
ej1.m
1  % EJEMPLO DE USO DE PRINTF
2
3  Re = 1500
4
5  fprintf('El numero de Reynold es:\n',Re)
6
7  Re = 13000
8
9  fprintf('El numero de Reynold es:\n',Re)
10
11
```

```
octave-3.2.4.exe:9> ej1
Re = 1500
El numero de Reynold es:
Re = 13000
El numero de Reynold es:
octave-3.2.4.exe:10>
```

HAY SALTO DE LINEA Y “octave 3.2.4>” SE MUESTRA EN LA SIGUIENTE LÍNEA

FPRINTF

LOS DATOS SE PIDEN AL USUARIO, SE MUESTRA POR PANTALLA (NO PONER ;) Y CON FORMATO CIENTIFICO (%e\n)

```
ej2.m
1  EJEMPLO DE USO DE FPRINTF
2
3  a = input ('Introduce un primer numero:')
4
5  b = input ('Introduce un segundo numero:')
6
7  c = a/b;
8
9  fprintf('El resultado de la division es: %e\n',c);
10
```

```
octave-3.2.4.exe:11> ej2
Introduce un primer numero:5800
a = 5800
Introduce un segundo numero:25000
b = 25000
El resultado de la division es: 2.320000e-001
octave-3.2.4.exe:15>
```

LOS DATOS SE PIDEN AL USUARIO, NO SE MUESTRA POR PANTALLA (PONER ;) Y CON FORMATO CIENTIFICO (%e\n)

```
ej2.m
1  EJEMPLO DE USO DE FPRINTF
2
3  a = input ('Introduce un primer numero:');
4
5  b = input ('Introduce un segundo numero:');
6
7  c = a/b;
8
9  fprintf('El resultado de la division es: %e\n',c);
10
```

```
octave-3.2.4.exe:15>
octave-3.2.4.exe:15> ej2
Introduce un primer numero:45000
Introduce un segundo numero:56900
El resultado de la division es: 7.908612e-001
octave-3.2.4.exe:16>
```


FPRINTF

SE PUEDE CONTROLAR EL NÚMERO DE DECIMALES. POR EJEMPLO, CON 0.2%f PROPORCIONA DOS CIFRAS DECIMALES. CON LA CIFRA ANTERIOR AL PUNTO INDICAMOS EL ESPACIO RESERVADO PARA EL RESULTADO POR PANTALLA

```
ej2.m
1  % EJEMPLO DE USO DE FPRINTF
2
3  a = input ('Introduce un primer numero:');
4
5  b = input ('Introduce un segundo numero:');
6
7  c = a/b;
8
9  fprintf('El resultado de la division es: %0.2f\n',c);
10
11 fprintf('El resultado de la division es: %8.2f\n',c);
12
13 fprintf('El resultado de la division es: %0.2e\n',c);
14
15 fprintf('El resultado de la division es: %8.2e\n',c);
16
17
```

```
octave-3.2.4.exe:16> ej2
Introduce un primer numero:5860
Introduce un segundo numero:8985114
El resultado de la division es: 0.00
El resultado de la division es:    0.00
El resultado de la division es: 6.52e-004
El resultado de la division es: 6.52e-004
octave-3.2.4.exe:17>
```


Ejemplo 2 (fprintf): Escribir las instrucciones en Octave adecuadas para que por pantalla se muestren las siguientes frases en el formato que se indica:

$a = 5$, $b = 48.56$, $c = -4.7864$, $d = 1111111111$

- | | |
|-----------------------------|--------------------------------|
| a) El valor de a es 5 | b) El valor de a es 5.00 |
| c) El valor de b es 49 | d) El valor de b es 48.56 |
| e) El valor de b es 48.5600 | f) El valor de c es -4.7864 |
| g) El valor de c es -4.8 | h) El valor de d es 1.111e+010 |
| i) El valor de d es 1.e+010 | |

```
octave-3.2.4.exe:2>
octave-3.2.4.exe:2> Ejemplo2_T5
El valor de a es 5
El valor de a es 5.00
El valor de b es 49
El valor de b es 48.56
El valor de b es 48.56000
El valor de c es -4.7864
El valor de c es -4.8
El valor de d es 1.111e+010
El valor de d es 1e+010
octave-3.2.4.exe:3> _
```

```

Ejemplo2_T5.m
1  a=5;
2  b=48.56;
3  c=-4.7864;
4  d=1.1111111111;
5  fprintf('El valor de a es %f\n',a);
6
7  fprintf('El valor de a es %2f\n',a);
8
9  fprintf('El valor de b es %f\n',b);
10
11 fprintf('El valor de b es %2f\n',b);
12
13 fprintf('El valor de b es %5f\n',b);
14
15 fprintf('El valor de c es %4f\n',c);
16
17 fprintf('El valor de c es %1f\n',c);
18
19 fprintf('El valor de d es %3e\n',d);
20
21 fprintf('El valor de d es %0e\n',d);
22
23

```

INTRODUCIMOS LOS VALORES DE a, b, c y d

```

octave-3.2.4.exe:2>
octave-3.2.4.exe:2> Ejemplo2_T5
El valor de a es 5
El valor de a es 5.00
El valor de b es 49
El valor de b es 48.56
El valor de b es 48.56000
El valor de c es -4.7864
El valor de c es -4.8
El valor de d es 1.111e+010
El valor de d es 1e+010
octave-3.2.4.exe:3> _

```

CAMBIAMOS EL FORMATO SE SALIDA DE LOS RESULTADOS CON FPRINTF

Ejemplo 3: Escribe programa *pmol* en Octave para el cálculo del peso molecular de moléculas orgánicas. Pide al usuario el número de átomos de carbono, hidrógeno y oxígeno. Calcula la masa a partir del peso molecular y muestra por pantalla el peso de la molécula.

```

1  % PROGRAMA PARA EL CÁLCULO DE PESO MOLECULAR
2
3  disp ('Programa para calculo peso molecular');
4
5  peso_C = input ('Introduce el numero de C:');
6
7  peso_H = input ('Introduce el numero de H:');
8
9  peso_O = input ('Introduce el numero de O:');
10
11  peso_molecular=peso_C*12+peso_H*1+peso_O*16;
12
13  fprintf ('El peso molecular es: %.2f g/mol \n', peso_molecular)
14

```

→ disp MUESTRA MENSAJE POR PANTALLA SIN FORMATO

→ input PARA PEDIR DATOS AL USUARIO Y GUARDARLOS EN UNA VARIABLE

fprintf PARA DAR DATOS POR PANTALLA CON FORMATO ESPECÍFICO

```

octave-3.2.4.exe:27
octave-3.2.4.exe:2>
octave-3.2.4.exe:2> Ejemplo3_T5
Programa para calculo peso molecular
Introduce el numero de C:12
Introduce el numero de H:14
Introduce el numero de O:254
El peso molecular es: 4222.00 g/mol-1
octave-3.2.4.exe:3>

```

**FPRINTF**

SE EMPLEA TAMBIEN CON MATRICES. EL COMANDO FPRINTF SE REPITE HASTA QUE SE HAYA REALIZADO CON TODOS LOS COMPONENTES DE LA MATRIZ

Ejemplo: x = 1:5

*fprintf ('%0.2f \n', x) devuelve por pantalla: 1.00
2.00
3.00*

...

SI LA VARIABLE ES UNA MATRIZ BIDIMENSIONAL. FPRINTF USA LOS VALORES DE UNA COLUMNA A LA VEZ, LUEGO PASA A LA SEGUNDA, ETC. CUIDADO AL USARLO (TENER ESTO EN CUENTA)

Ejemplo: a = 1:3

*b = a.*12*

*table = [a;b] devuelve por pantalla: table = 1 2 3
12 24 36*

*fprintf ('%4.0f %7.2f \n', table) devuelve por pantalla: 1 12.00
2 24.00
3 36.00*

*fprintf ('%4.0f %7.2f \n', a, b) devuelve por pantalla: 1 2.00
3 12.00
24 36.00*

FPRINTF

SI QUEREMOS PONER UN % TENEMOS QUE PONERLO DOS VECES, SINO LO HACE, LO INTERPRETA COMO EL FORMATO DE LA VARIABLE A DAR POR PANTALLA

Ejemplo:

fprintf ('La tasa de interés es %5.2f %% \n', 5)

devuelve por pantalla: La tasa de interés es 5.00. %

Ejemplo 4a. (fprintf y salida con formato): Analizar el comportamiento de un objeto en caída libre. La ecuación que lo describe es: $d = (1/2) \cdot g \cdot t^2$. En vez de hacer una gráfica, hacer una tabla con los resultados y usar los comandos *disp* y *fprintf*.

```

1  % RESOLUCION EJEMPLO 1 TEMA 5
2  % DETERMINAMOS LA DISTANCIA EN CAIDA LIBRE
3  % EL USUARIO PROPORCIONA LA GRAVEDAD Y UN VECTOR DE TIEMPO
4
5
6  clear, clc
7
8  g = input('¿Cual es el valor de la aceleracion debido a la gravedad?')
9
10 inicio = input ('¿Que tiempo de inicial quiera?')
11
12 final = input ('¿Que tiempo final?')
13
14 incremento = input ('¿y el incremento?')
15
16 % CREAMOS EL VECTOR
17 t=[inicio:incremento:final]
18
19 % CALCULAMOS LA DISTANCIA
20 d = (1/2)*g*t.^2
21
22 % CREAMOS UNA MATRIZ CON LOS RESULTADOS
23 resultados = [t;d]
24
25
26 % MOSTRAMOS LOS RESULTADOS POR PANTALLA
27
28 fprintf('Para una aceleracion debida a la gravedad de %5.1f segundos se calcularon los siguientes datos \n',g)
29
30 disp('Distancia recorrida en cada libre')
31
32 % ESTA SERIA LA CABECERA DE LA TABLA
33 disp('t (s)      distancia (m)')
34
35 fprintf('%8.0f %10.2f\n',resultados)
36

```

PARA BORRAR LA PANTALLA DE OCTAVE

% CREAMOS UNA MATRIZ CON LOS RESULTADOS

resultados = [t;d]

disp('t (s) distancia (m)')

fprintf('%8.0f %10.2f\n',resultados)

Ejemplo 4b. (fprintf y salida con formato): Analizar el comportamiento de un objeto en caída libre. La ecuación que lo describe es: $d = (1/2) \cdot g \cdot t^2$. en vez de hacer una gráfica, hacer una tabla con los resultados y usar los comandos *disp* y *fprintf*.

```

¿Cual es el valor de la aceleracion debido a la gravedad?1.6
g = 1.6000
¿Que tiempo de inicial quiere?0
inicio = 0
¿Que tiempo final?100
final = 100
¿y el incremento?10_

incremento = 10
t =

    0    10    20    30    40    50    60    70    80    90   100

d =

    0     80    320    720   1280   2000   2880   3920   5120   6480   8000

resultados =

    0     10     20     30     40     50     60     70     80     90    100
    0     80    320    720   1280   2000   2880   3920   5120   6480   8000

Para una aceleracion debida a la gravedad de 1.6 segundos se calcularon los si
guientes datos
Distancia recorrida en caída libre
t (s)    distancia (m)
    0         0.00
    10        80.00
    20       320.00
    30       720.00
    40      1280.00
    50      2000.00
lines 1-23 -- <f>orward, <b>ack, <q>uit

```

Clear +clc limpia la
ventana de
comandos

FORWARD (VER DATOS POSTERIORES)

BACK (VER DATOS ANTERIORES)

**QUIT (SALIMOS DE LA VISIÓN DE DATOS,
PERO ESTAN EN LA MEMORIA DE
OCTAVE)**

Ejemplo 5a. Algunas reacciones transcurren mediante la formación de un intermedio, como en las reacciones consecutivas de primer orden $A \rightarrow B \rightarrow C$. Las ecuaciones que rigen estas reacciones, y que proporcionan la concentración de cada especie en función del tiempo son las siguientes:

$$\begin{aligned}[A] &= [A]_0 e^{-k_1 t} \\ [B] &= k_1 [A]_0 \left\{ \frac{e^{-k_1 t} - e^{-k_2 t}}{k_2 - k_1} \right\} \\ [C] &= [A]_0 \left\{ 1 + \left(\frac{k_1 e^{-k_2 t} - k_2 e^{-k_1 t}}{k_2 - k_1} \right) \right\}\end{aligned}$$

Donde $[A]$, $[B]$ y $[C]$ son las concentraciones de las respectivas especies, $[A]_0$ la concentración inicial de A y k_1 y k_2 las constantes de velocidad.

Escribe un programa que pida al usuario los valores de $[A]_0$, k_1 y k_2 , calcule las concentraciones de las tres especies en función del tiempo.

Ejemplo 5b.

- Podemos resolver el problema calculando las concentraciones de A, B y C a partir de las ecuaciones. Sin embargo, ¿hasta que valor de tiempo debemos calcularlo?
- Según la secuencia de reacciones ($A \rightarrow B \rightarrow C$), el reactivo A va a desaparecer. Suponemos que va a quedar un 1% y calculamos el tiempo máximo de reacción:

$$[A] = 0.01 * [A]_0 \rightarrow 0.01 * [A]_0 = [A]_0 * \exp(-k_1 * t_{\max}) \quad \Rightarrow \quad [A] = [A]_0 e^{-k_1 t}$$

- Calculamos ese tiempo máximo con el comando round (redondea al valor más cercano) y creamos el vector de tiempos con el comando linspace (t inicial, t final, numero de valores).
- Damos al usuarios los resultados por pantalla.

Ejemplo 5c.

```

Ejemplo5_T5.m
1
2 % PROGRAMA PARA CALCULAR LA VARIACION CON EL TIEMPO DE LA CONC. A, B Y C
3
4 % REACCION CONSECUTIVA A->B->C CON CONSTANTES CINETICAS K1 Y K2
5
6 % PEDIMOS AL USUARIO LOS VALORES DE A0, K1 Y K2
7
8 A0=input('Introduzca el valor de concentracion inicial de A:');
9
10 k1=input('Introduzca el valor de la constante k1:');
11
12 k2=input('Introduzca el valor de la constante k2:');
13
14 % CALCULAMOS EL TIEMPO MÁXIMO Y EL VECTOR DE TIEMPOS
15
16 tmax=round(-log(0.01)/k1);
17
18 t=linspace(0,tmax,100);
19
20 % CALCULAMOS LOS VALORES DE LAS CONCENTRACIONES DE A,B Y C
21
22 A=A0*exp(-k1*t);
23
24 B=k1*A0*(exp(-k1*t)-exp(-k2*t))/(k2-k1);
25
26 C=A0*(1+(k1*exp(-k2*t)-exp(-k1*t))/(k2-k1));
27
28 % IMPRESION DE RESULTADOS
29
30 disp('Las concentraciones de las tres especies A,B y C son:')
31
32 disp('t A B C') %LAS CABECERAS DE LAS COLUMNAS DE RESULTADOS
33
34 disp([t' A' B' C']) %TRANSPONEMOS LOS RESULTADOS PARA QUE SEA MAS VISUAL

```

EL USUARIO
INTRODUCE LOS
VALORES DE
CONC.INICIAL Y
CONSTANTES

CALCULAMOS EL TIEMPO MÁXIMO Y
GENERAMOS EL VECTOR DE TIEMPOS

INTRODUCCIMOS
LAS ECUACIONES

GENERAMOS UNA
MATRIZ
TRANSPONIENDO
LOS VECTORES DE
RESULTADOS

Ejemplo 5d.

```
octave-3.2.4.exe:25> Ejemplo5_T5
Introduzca el valor de concentracion inicial de A:50
Introduzca el valor de la constante k1:0.6
Introduzca el valor de la constante k2:0.3
octave-3.2.4.exe:26>
```

CABECERA DE LA
COLUMNA

MATRIZ DE
RESULTADOS

```
Las concentraciones de las tres especies A,B y C son:
t A B C
```

0.0000e+000	5.0000e+001	-0.0000e+000	1.1667e+002
8.0808e-002	5.2484e+001	2.3377e+000	1.1507e+002
1.6162e-001	5.5091e+001	4.5088e+000	1.1351e+002
2.4242e-001	5.7828e+001	6.5225e+000	1.1199e+002
3.2323e-001	6.0701e+001	8.3876e+000	1.1051e+002
4.0404e-001	6.3717e+001	1.0112e+001	1.0906e+002
4.8485e-001	6.6882e+001	1.1705e+001	1.0764e+002
5.6566e-001	7.0205e+001	1.3172e+001	1.0626e+002
6.4646e-001	7.3693e+001	1.4521e+001	1.0491e+002
7.2727e-001	7.7354e+001	1.5760e+001	1.0360e+002
8.0808e-001	8.1196e+001	1.6893e+001	1.0231e+002
8.8889e-001	8.5230e+001	1.7928e+001	1.0106e+002
9.6970e-001	8.9464e+001	1.8870e+001	9.9839e+001
1.0505e+000	9.3909e+001	1.9725e+001	9.8645e+001
1.1313e+000	9.8574e+001	2.0497e+001	9.7480e+001
1.2121e+000	1.0347e+002	2.1192e+001	9.6343e+001
1.2929e+000	1.0861e+002	2.1814e+001	9.5233e+001
1.3737e+000	1.1401e+002	2.2368e+001	9.4150e+001
1.4545e+000	1.1967e+002	2.2857e+001	9.3092e+001
1.5354e+000	1.2562e+002	2.3286e+001	9.2060e+001
1.6162e+000	1.3186e+002	2.3659e+001	9.1053e+001
1.6970e+000	1.3841e+002	2.3979e+001	9.0069e+001

```
lines 1-24 -- (f)orward, (b)ack, (q)uit
```

USO DE VARIABLES

clear



PARA BORRAR LAS VARIABLES ALMACENADAS, INTRODUCIDAS POR EL USUARIO (EVITA ERRORES). ESCRIBIR CLEAR AL COMENZAR UN CONJUNTO DE ORDENES O COMANDOS.

clc: borra la ventana de comandos de octave

who



MUESTRA LAS VARIABLES DEFINIDAS EN LA HOJA DE TRABAJO ACTUAL.

**isvarname
nombre_variable**



PARA VERIFICAR QUE EL NOMBRE DE LA VARIABLE ESTA PERMITIDO. SI LA RESPUESTA ES 1 ES VALIDO, SI ES 0 NO LO ES.

```
octave-3.2.4.exe:4>
octave-3.2.4.exe:4> isvarname time
ans = 1
octave-3.2.4.exe:5> isvarname 55
ans = 0
octave-3.2.4.exe:6> isvarname porcentaje%
ans = 0
octave-3.2.4.exe:7> isvarname _55
ans = 1
octave-3.2.4.exe:8>
```

USO DE VARIABLES

iskeyword



PROPORCIONA UNA LISTA CON LAS PALABRAS CLAVE QUE USA EL PROGRAMA QUE NO PUEDEN SER UTILIZADAS PARA NOMBRAR A OTRA VARIABLE

```
ans =
(
[1,1] = _FILE_
[2,1] = _LINE_
[3,1] = break
[4,1] = case
[5,1] = catch
[6,1] = continue
[7,1] = do
[8,1] = else
[9,1] = elseif
[10,1] = end
[11,1] = end_try_catch
[12,1] = end_unwind_protect
[13,1] = endfor
[14,1] = endfunction
[15,1] = endif
[16,1] = endswitch
[17,1] = endwhile
[18,1] = for
[19,1] = function
lines 1-22 — <f>orward, <b>ack, <q>uit
```

```
[12,1] = end_unwind_protect
[13,1] = endfor
[14,1] = endfunction
[15,1] = endif
[16,1] = endswitch
[17,1] = endwhile
[18,1] = for
[19,1] = function
[20,1] = global
[21,1] = if
[22,1] = otherwise
[23,1] = persistent
[24,1] = return
[25,1] = static
[26,1] = switch
[27,1] = try
[28,1] = until
[29,1] = unwind_protect
[30,1] = unwind_protect_cleanup
[31,1] = while
```

FORWARD (VER DATOS POSTERIORES)

BACK (VER DATOS ANTERIORES)

QUIT (SALIMOS DE LA VISIÓN DE DATOS, PERO ESTAN EN LA MEMORIA DE OCTAVE)

```
lines 15-36/36 <END> — <f>orward, <b>ack, <q>uit
```

GESTIÓN DE FICHEROS DE HOJAS DE CÁLCULO

csvread



PARA IMPORTAR FICHEROS .csv A OCTAVE

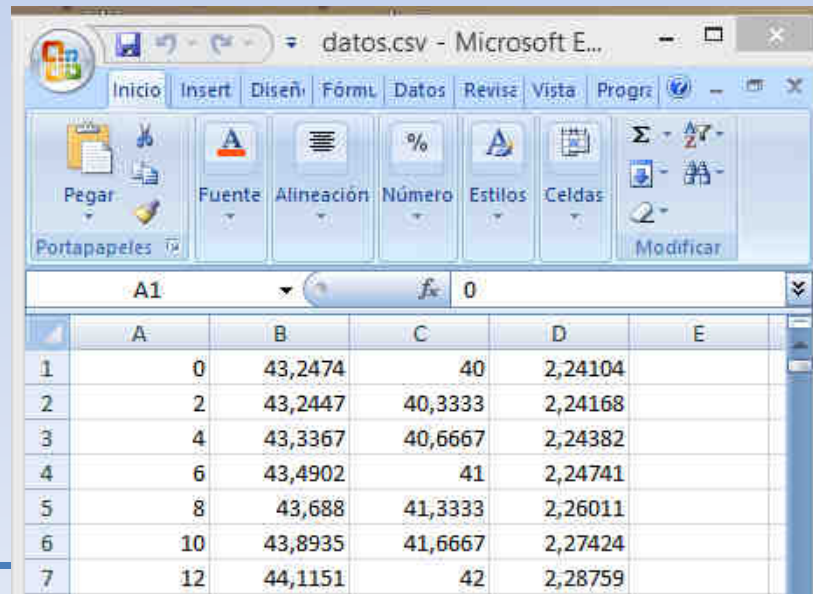
`csvread ('nombre_fichero.csv')`

PARA IMPORTAR FICHEROS .csv A OCTAVE Y ASIGNARSELOS A UNA VARIABLE

`A = csvread ('nombre_fichero.csv')`

SI NO SABEMOS EL TAMAÑO DE LOS DATOS IMPORTADOS USAR COMANDO `size`

`size (A)`



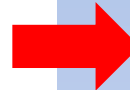
	A	B	C	D	E
1	0	43,2474	40	2,24104	
2	2	43,2447	40,3333	2,24168	
3	4	43,3367	40,6667	2,24382	
4	6	43,4902	41	2,24741	
5	8	43,688	41,3333	2,26011	
6	10	43,8935	41,6667	2,27424	
7	12	44,1151	42	2,28759	

csvread

```

1
2 % IMPORTAMOS LOS DATOS DEL ARCHIVO datos.csv
3
4 csvread('datos.csv')
5
6 % ASIGNAMOS UN NOMBRE A LOS DATOS
7
8 A=csvread('datos.csv')
9
10 % PEDIMOS EL TAMAÑO DE LA MATRIZ DE DATOS
11
12 size(A)
13
14

```



```

0      2474      24104      0
2      2447      3333      24168
4      3367      6667      24382
6      4902      24741      0
8      688      3333      26011
10     8935      6667      27424
12     1151     28759      0
14     3639      3333     29932
16      625      6667     30826
18     8846     31589      0
20     1424      3333     32148
22     4042      6667     32628
24     6793     32986      0
26     9562      3333     33297
28     2407      6667     33535
30     5269      3375      0
32      805      3333     33942
34      928      6667     34093
36     3815     34243      0
38      672      3333     3437
40     2327      8887     33773
42      242     34603      0
lines 1-24 -- (f)orward, (b)ack, (q)uit

```

```

A =
0      2474      24104      0
2      2447      3333      24168
4      3367      6667      24382
6      4902      24741      0
8      688      3333      26011
10     8935      6667      27424
12     1151     28759      0
14     3639      3333     29932
16      625      6667     30826
18     8846     31589      0
20     1424      3333     32148
lines 2274-3297 -- (f)orward, (b)ack, (q)uit

```

```

octave-3.2.4.exe:37
octave-3.2.4.exe:3> size(A)
ans =

```

```

2281      4

```


csvread

ES POSIBLE QUE NO QUERAMOS IMPORTAR TODOS LOS DATOS DEL FICHERO .csv (LAS PRIMERAS FILAS CORRESPONDEN AL TÍTULO DE LAS COLUMNAS). PARA ELLO FIJAMOS A PARTIR DE QUE FILA Y COLUMNA QUEREMOS EXPORTAR.

Ejemplo: Supongamos que queremos importar sólo las entradas del fichero “datos.csv” que se encuentran por debajo de la primera fila y a al derecha de la segunda columna:

```
csvread ('datos.csv',1,2)
```

- Si queremos excluir los datos de las dos primeras columnas:

```
csvread ('datos.csv',0,2)
```

- Si queremos importar sólo los datos que hay entre la segunda y la cuarta fila (ambas inclusive) y la primera y la tercera columna (ambas inclusive):

```
csvread ('datos.csv',[1,0,3,2])
```


GESTIÓN DE FICHEROS DE HOJAS DE CÁLCULO

cswrite



PARA EXPORTAR DE OCTAVE A UN FICHERO .csv

EJEMPLO: PRIMERO GENERAMOS UNA MATRIZ ALEATORIA CON LAS FILAS Y LAS COLUMNAS NECESARIAS

`A = rand(3,4)`

GUARDAMOS LOS DATOS DE A EN UN FICHERO AL QUE LLAMAMOS: datosA.csv

`csvwrite('datosA.csv',A)`

```

1
2
3  % GENERAMOS LA MATRIZ ALEATORIA
4
5  A=rand(3,4)
6
7  % EXPORTAMOS LOS DATOS A UN FICHERO .csv
8
9  csvwrite('datosA.csv',A)
10
11
    
```



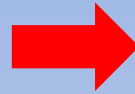
```

octave-3.2.4.exe:4>
octave-3.2.4.exe:4> Exportar_datos
A =

    0.107146    0.500540    0.896652    0.825985
    0.745571    0.273640    0.624229    0.615089
    0.724847    0.043966    0.594897    0.693480
    
```

SE GENERA ARCHIVO datos.csv EN LA CARPETA ELEGIDA

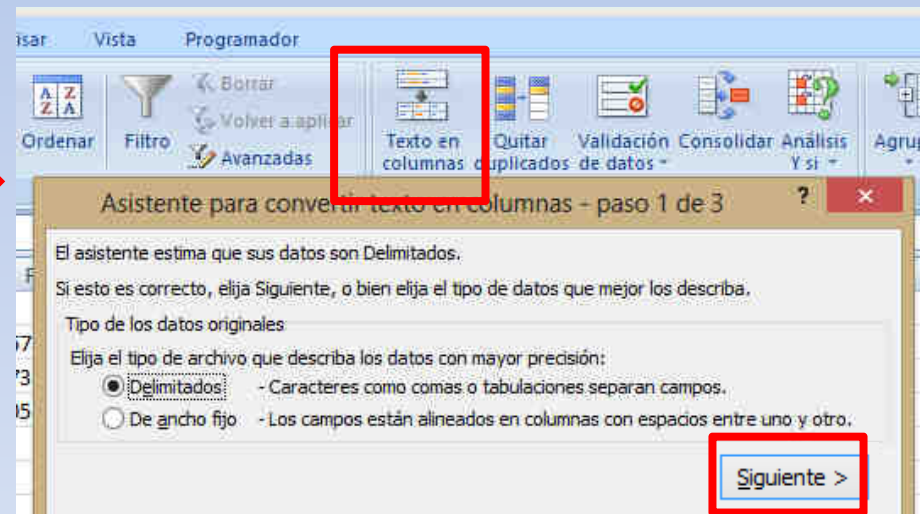
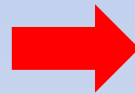
ARCHIVO GENERADO



	A	B	C	D	E
1	0.1071460943807368,	0.5005397161452407,	0.8966524884574461,	0.82	
2	0.7455712647467564,	0.2736403371315679,	0.6242294418910955,	0.61	
3	0.7248474494172394,	0.04396604007123177,	0.594896914998516,	0.69	
4					
5					

PODEMOS EXPORTAR LOS DATOS DEJANDO FILAS O COLUMNAS VACÍAS ARRIBA O A LA IZQUIERDA, DE TAL MODO QUE PODAMOS EDITAR EL FICHERO
`csvwrite('datosA2.csv',A,1,1)`

	A	B	C	D
1				
2	0.9717054717624791,	0.6043645655802032,	0.12920341780	
3	0.02195481173999385,	0.8437901039260409,	0.1238882130	
4	0.04304175669713332,	0.5780247248600344,	0.8984622280	
5				



Separadores

☐ Tabulación

☐ Punto y coma

☒ Coma

☐ Espacio

☐ Otro:



	A	B	C	D
1				
2		0.9717054717624791	0.6043645655802032	0.12920341780
3		0.02195481173999385	0.8437901039260409	0.1238882130
4		0.04304175669713332	0.5780247248600344	0.8984622280
5				

**SE GENERA
UNA FILA
NUEVA Y UNA
COLUMNA
VACÍAS**

GESTIÓN DE FICHEROS DE HOJAS DE CÁLCULO

dlmwrite

dlmread



PARA IMPORTAR/EXPORTAR DE FICHEROS .txt A OCTAVE Y DE OCTAVE A UN ARCHIVO .txt, RESPECTIVAMENTE. EN ESTE CASO ELEGIMOS UN TABULADOR ESPECÍFICO (":" ";" " " " ")

Ejemplo:

- Generamos matriz de cinco filas y tres columnas: $B = \text{rand}(5,3)$
- Guardamos los datos de B en un fichero .txt separando las columnas por puntas y comas (exportamos los datos):

dlmwrite('datosB.txt',B, ';')

- Podemos ver el contenido del fichero creado abriéndolo con cualquier editor de textos y recuperar su contenido en Octave con la función dlmread especificando el delimitador usado (importamos los datos)

A=dlmread('datosB.txt' , ';')

dlmread

dlmwrite

```

1
2
3  * GENERAMOS LA MATRIZ ALEATORIA
4
5  B=rand(5,3)
6
7  * EXPORTAMOS LOS DATOS A UN FICHERO .CSV
8
9  dlmwrite('datosB.csv',B,':');
10
11

```



```

octave-3.2.4.exe:2>
octave-3.2.4.exe:2> Exportar_datos2
B =
    0.733462    0.865755    0.824542
    0.635283    0.857949    0.189507
    0.590912    0.169789    0.703893
    0.055395    0.976616    0.011292
    0.582020    0.529030    0.692552

```



datosB.txt: Bloc de notas

```

Archivo  Edición  Formato  Ver  Ayuda
0.733461724533513;0.8657551846488332;0.8245418671632024
0.6352828780508132;0.8579491292439283;0.1895066179006518
0.5909123048956669;0.1697894563799657;0.7038933134502232
0.05539513957904223;0.9766156320731152;0.01129174686247152
0.5820198426996535;0.5290303155152881;0.6925519647354623

```