



Objetivos:

Al finalizar la práctica, el alumno deberá ser capaz de:

- Conocer el funcionamiento básico del programa Matlab.
- Crear funciones/módulos específicos mediante ficheros M.
- Programar estructuras de control selectivas y repetitivas en Matlab.
- Trabajar con tipos de datos estructurados en Matlab.
- Acceder a ficheros de datos desde Matlab.

Modo de realización de la práctica:

- Esta práctica se plantea como un caso real de aprendizaje autónomo de un nuevo lenguaje/herramienta de programación científica a partir de los conocimientos que el alumno tiene sobre otros lenguajes (p. ej. C/C++).
- El guión presenta un conjunto de ejercicios, para cuya realización el alumno deberá buscar información de manera libre.
- Se proporciona un documento básico de aprendizaje así como una orientación sobre los apartados donde se encontrará la información de interés.
- No obstante, animamos al alumno a buscar sus propias fuentes para emular la situación a la que se enfrentará en la vida real.

Ejercicios:

1. Familiarizaros con el entorno de Matlab (apartados 2.3 y 2.5) y responded a las siguientes preguntas:
 - ¿Qué es Matlab y para qué se puede utilizar?
 - ¿Cómo se usa la ventana de comandos? ¿Cómo podemos repetir comandos ya ejecutados?
 - ¿Qué es el espacio de trabajo y para qué se puede utilizar?
 - ¿Qué es el directorio actual y cómo se puede modificar?
 - ¿Qué son los ficheros M y cómo se pueden crear?
 - ¿Cómo se usa la herramienta Help?
 - Utilizando el comando *help*, describid como se realiza la entrada/salida de datos con *input* y *disp*.
2. Escribid el siguiente programa dentro del fichero *primero.m* y comprended su funcionamiento:

```
a = input('hola, escribe un nombre: ');  
a+a = a;  
disp(a);
```

¿Qué hace el programa? ¿Hay algún error? ¿Cuál?
3. **FAHRENHEIT:** Haced un programa que convierta una temperatura de grados Celsius a Grados Fahrenheit. El alumno introducirá la temperatura en grados Celsius y el ordenador la mostrará por pantalla en grados Fahrenheit de acuerdo con la fórmula siguiente: $F = 32 + C \times 9/5$.
4. **MÁXIMO:** Haced un programa que pida tres números enteros por teclado y muestre por pantalla el valor del mayor. (Usad la sentencia *if* descrita en el apartado 6.1.1).
5. **PAR:** Haced un programa que permita introducir un número entero por teclado y diga si es par o impar. Utilizad la función *mod* (más detalles en *help*) para calcular el resto de la división entre dos números. Esto es, *mod(x,2)* valdrá 0 si x es par y 1 si es impar.
6. **CALCULADORA:** Haced un programa que pida dos números enteros y muestre el siguiente menú:
 - 1.- Sumar
 - 2.- Restar



- 3.- Multiplicar
4.- Dividir

El usuario escogerá la opción deseada y el programa mostrará por pantalla el resultado de la operación. Si la opción elegida no existe, el programa mostrará un mensaje informativo del error. (Utilizad la sentencia *switch* descrita en el apartado 6.1.2).

7. **SERIE:** Escribid un programa que calcule, para dos valores n y i introducidos por teclado, el valor de la serie siguiente: $\sum_{i=0}^n \frac{i}{2}$. Haced dos versiones diferentes del programa: una que use un bucle *for* (apartado 6.1.3) y la otra un bucle *while* (apartado 6.1.4).

8. **TABLAMAX:** El siguiente programa pide dos números enteros y debería mostrar por pantalla la tabla de multiplicar del mayor. El programa utiliza las funciones de Matlab (apartado 6.3.2). Cada función en Matlab se aloja en un fichero M con el mismo nombre de la función. Desgraciadamente, el programador ha cometido un error y el programa no hace lo que se espera. Introducid el siguiente código, corregid los errores y tratad de comprender el funcionamiento del programa:

```
function res = maximo(x, y)
    if (a > b)
        res = a;
    else
        res = b;
    end
end

function tablamult(x)
    for i=1:10
        fprintf('%d per %d = %d\n',x,i,x*i);
    end
end

% Programa principal
a = input('Introduce un número entero positivo');
b = input('Introduce otro número entero positivo');
resul = maximo(a,b);
tablamult(resul);
```

9. **MISCELANIA:** De acuerdo con lo que habéis aprendido en el ejercicio anterior, escribid las funciones que permiten ejecutar el código siguiente:

```
x = input('Escribe un número: ');
y = input('Escribe otro número: ');
z = input('Escribe un tercero: ');

if ispar(x)
    fprintf('El primer número es par\n');
else
    fprintf('El primer número es impar\n');
end

med = media(x, y, z);
fprintf('La media de los tres números es: %.2f\n', med);

[max,min] = maximo_minimo(x, y, z);
fprintf('El máximo de los tres números es: %d\n', max);
fprintf('El mínimo de los tres números es: %d\n', min);
```

10. **VECTORES:** Haced un programa que pida números por teclado hasta que se introduzca el valor 0. Los valores introducidos rellenarán dos vectores (apartados 3 y 3.1), uno de números pares y otro de impares. A continuación, mostrará por pantalla ambos vectores según se especifica en el siguiente ejemplo:
Escribe un número entero: 2



```
Escribe un número entero: 4  
Escribe un número entero: 3  
Escribe un número entero: 5  
Escribe un número entero: 7  
Escribe un número entero: 0
```

```
Vector de pares: 2 elementos  
posición 1: 2  
posición 2: 4
```

```
Vector de impares: 3 elementos  
posición 1: 3  
posición 2: 5  
posición 3: 7
```

11. **IDENTIDAD:** Se define la matriz identidad de orden n como una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal, que son iguales a 1. Por ejemplo, la matriz identidad de orden 3 es:

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Haced una función llamada `genera_identidad` que reciba el orden de la matriz y devuelva la matriz identidad de dicho orden.

12. **PUNTO:** Un punto es un elemento formado por dos coordenadas x e y . Haced un programa que pida dos puntos, los guarde en dos variables de tipo registro (apartado 5.3), calcule y muestre la distancia entre ellos según la fórmula siguiente:

$$\text{Distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

NOTA: Cread la función `calcula_distancia` que reciba dos puntos (registros con los campos x e y) y devuelva la distancia.

13. **TRABAJADORES:** El fichero “sueldos.dat” del Aula Virtual contiene información de los sueldos mensuales de los trabajadores de una empresa. Los datos siguen el formato (Nombre, Sueldo, Antigüedad) que se muestra a continuación:

```
Juan 1800 3  
Alberto 2100 12  
Ana 1500 7
```

Haced un programa que lea el fichero anterior (apartado 6.6) e informe sobre:

- ¿Cuántos trabajadores tiene la fábrica?
- ¿Cuál es el nombre y la antigüedad de los trabajadores que cobran más de 2000€?