

PLANTILLA DE UN PROGRAMA

```
% Programa Nombre
% Entradas: declaración, restricciones, ...
% Salidas: descripción
% Variables: identificador: tipo
               <Cuerpo del programa. Lista de instrucciones>
```

PLANTILLA PARA MÓDULOS

```
function [<salidas> =] <nombre> ( [ <parám. entrada> ] )
% Descripción: que hace la función
% Parámetros de entrada:
    identificador: tipo y descripción
% Parámetros de salida:
    descripción | var1: tipo y descripción , var2: tipo y desc.
% Variables (internas):
    identificador: tipo

    <lista de instrucciones>
    < salidas = valores (en algún punto del programa) >
```

Llamadas a módulos

```
[salidas =] nombre_de_la_función( parámetros_de_entrada )
```

Ejemplos de llamadas:

cont = Ejemplo1Funcion (y)	llamada con 1 parámetro de entrada y 1 de salida
disp (Ejemplo2Funcion(3, 2))	llamada con 2 parámetros de entrada y 1 de salida
[cont,cont2] = Ejemplo3Funcion (a)	llamada con 1 parámetro de entrada y 2 de salida
OtroEjemploProcedim(x)	llamada con un parámetro de entrada y sin parámetro de salida
cont= EjemploFuncion	llamada sin parámetros de entrada y 1 de salida

Operaciones secuenciales

Asignación:

```
<variable> = <expresión>
```

Entrada/Lectura:

```
<variable>= input ('<texto>')
```

Salida/Escritura

```
disp ('texto')
disp (<expresión>)
disp([<lista de valores>])
    %num2str(<identificador>) para
    %ver los valores de las variables
```

Estructura Condicional

Selección Simple

```
if <condición>
    <instrucciones>
else
    <instrucciones>
end
```

Selección Múltiple

```
switch <expresión>
    [case <valor 1>
        <instrucciones>]
    ...
    [case {<valor n1>, <valor n2>, ...}
        <instrucciones>]
    [otherwise
        <instrucciones>]
end
```

Estructuras Cíclicas

Ciclo Contado.

```
for <variable> = <lista>
    <instrucciones>
end
```

```
<lista>:
    inicio [: incremento ] : fin
    [valor1, valor2, ...]
```

Ciclo Condicional

```
while (<expresión lógica>)
    <Bloque instrucciones>
end
```

Operadores

Numéricos

+-	Signo
^	Potencia
*	Multiplicación
/	División
+	Suma
-	Resta
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que
==	Igual
~=	Distinto

Booleanos/Lógicos

~	Negación lógica, not(a)
&&	Y lógico, and(a,b)
	O lógico, or(a,b)

Con arrays y escalares

v+k	Suma
v-k	Resta
v*k	Multiplicación
v/k	Divide cada elemento de v por k
k./v	Divide k por cada elemento de v
v.^k	Potencia de cada elemento de v a k
k.^v	Potencia de k elevado a cada elemento de v

Útiles

%	Comentario
;	Evitar eco salida
help	Ayuda

Variables especiales

ans	Nombre por defecto de la variable usada para los resultados.
pi	Razón de una circunferencia a su diámetro.
eps	Número más pequeño tal que, cuando se le suma 1, crea un número en coma flotante mayor que 1.
inf	Infinito, por ejemplo 1/0.
NaN	Magnitud no numérica (Not a Number), por ejemplo, 0/0.
i y j	$i = j = \sqrt{-1}$
realmin	El número real positivo más pequeño que podemos utilizar.
realmax	El número real positivo más grande que podemos utilizar.

Nota:

- Los corchetes ([...]) se utilizan en la plantilla y en el resumen de instrucciones en Matlab para indicar la opcionalidad, es decir, indicar que estos elementos pueden o no aparecer en la sintaxis de la instrucción, dependiendo si se necesita esa funcionalidad. Las excepciones a esta norma son:
 - los arrays, que definen la lista de elementos utilizando corchetes, ej. [1,2,3] y
 - los módulos que necesitan devolver entre corchetes la lista de salidas, en el caso de que esta lista sea superior a 1. Ej. [f,c]=size(M)
- Las palabras y frases entre signos de mayor y menor (<...>) sirven para indicar expresiones NO interpretadas por Matlab.

RESUMEN DE INSTRUCCIONES DE MATLAB

Arrays	
Definición	
<code>v = [x,y,...]</code>	vector. 1 dimensión de filas
<code>m = [x;y;...]</code>	matriz. 1 dimensión de columnas
<code>m = [w,x; y,z]</code>	matriz de 2x2
<code>v = a[h:b]</code>	Lista de números (vector) desde a hasta b saltando de h en h

Acceso a los elementos	
<code><var>(<pos>)</code>	Array unidimensional. Acceso a la posición <pos> del array <var>. <i>Los índices comienzan en 1.</i>
<code><var>(<f>,<c>)</code>	Array bidimensional. Acceso a la fila <f>, columna <c> del array <var>
<code><var>(<f>,:) </code>	Acceso a toda la fila <f>
<code><var>(:,<c>)</code>	Acceso a toda la columna <c>

Funciones comunes	
<code>tam=length(v)</code>	Tamaño del vector
<code>linspace(a,b,n)</code>	Vector de n eltos. espaciados linealmente
<code>[f,c]=size(m)</code>	tamaño de una matriz
<code>ones(f,c)</code>	matriz de unos
<code>zeros(f,c)</code>	matriz de ceros
<code>rand(f,c)</code>	matriz de n° aleatorios
<code>eye(f,c)</code>	matriz identidad
<code>t'</code>	matriz transpuesta de t
<code>diag(v)</code>	matriz con el vector v como diagonal
<code>inv(a)</code>	invierte la matriz
<code>det(a)</code>	determinante de a
<code>+ -* ^ \</code>	Suma, resta, multiplicación, división

<code>. * , . ^</code> <code>. / , . \</code>	Multiplicación, potencia, división, elemento a elemento
<code>sum(v)</code>	Suma los elementos de v
<code>cross(v,w)</code>	Producto vectorial de w y v
<code>mean (v)</code>	Media de v
<code>mtimes(m1,m2)</code>	Multiplicación de matrices
<code>mrdivide(m1,m2)</code>	División de matrices
<code>mpow(m, y)</code>	Potencia de una matriz ^y
help matfun	Lista de funciones para trabajar con matrices.

Funciones comunes en Matlab	
<code>abs(x)</code>	Valor absoluto o magnitud de un número complejo.
<code>acos(x)</code>	Inversa del coseno.
<code>acosh(x)</code>	Inversa del coseno hiperbólico.
<code>angle (x)</code>	Ángulo de un número complejo.
<code>asin (x)</code>	Inversa del seno.
<code>asinh (x)</code>	Inversa del seno hiperbólico.
<code>atan (x)</code>	Inversa de la tangente.
<code>ceil(x)</code>	Redondea hacia más infinito.
<code>cos (x)</code>	Coseno.
<code>exp(x)</code>	Exponencial: e ^x
<code>factor (n)</code>	Lista de factores primos de n
<code>factorial(n)</code>	Factorial de n
<code>fix(x)</code>	Redondea hacia cero.
<code>floor(x)</code>	Redondea hacia menos infinito.

<code>gcd(x,y)</code>	Máximo común divisor de x e y, salida en un array
<code>idivide(x,y)</code>	División entera
<code>imag (x)</code>	Parte imaginara de un número complejo.
<code>lcm(x,y)</code>	Mínimo común múltiplo de x e y, salida en un array
<code>log(x)</code>	Logaritmo natural.
<code>log10(x)</code>	Logaritmo decimal.
<code>mod(x,y)</code>	Módulo de x dividido entre y
<code>real(x)</code>	Parte real de un número complejo.
<code>rem(x,y)</code>	Resto de de x/y.
<code>round(x)</code>	Redondea hacia el entero más próximo.
<code>sign(x)</code>	Devuelve el signo del argumento.
<code>sin(x)</code>	Seno.
<code>sqrt(x)</code>	Raíz cuadrada.
<code>tan(x)</code>	Tangente.

Archivos	
Abrir un archivo:	
<code><id>=fopen (<arch>,<modo>)</code>	
% id = -1 si hay un error. >2 en caso contrario	
Modos:	
<code>'w' , 'w+' </code>	Escritura, lectura y escritura.
<code>'r' , 'r+' </code>	Lectura, lectura y escritura. Debe existir previamente
<code>'a' , 'a+' </code>	añadir información, lectura y añadir.

Nota: Se añade una b para archivos binarios.

Cerrar un archivo:	
<code><error>=fclose (<id>)</code>	
% error = -1 si hay un error. 0 en caso contrario	

Escribir en un archivo de texto:	
<code><bytes>=fprintf([<id>],[<texto>],[<v1>,<v2>,...])</code>	
% <bytes>: número de bytes escritos % <texto>: en este valor se incluyen valores de formato, si se incluyen variables en el mensaje.	
Valores de Formato:	
<code>'%d' ' %i' </code>	Enteros
<code>'%f' ' %g' </code>	Reales
<code>'%c' </code>	Un solo carácter
<code>'%s' </code>	cadenas (solo lee o escribe hasta que encuentra un espacio, tab., o salto de línea)
<code>'\n' </code>	Salto de línea
<code>'\t' </code>	Tabulador

Leer de un archivo de texto:	
<code>[var,lee]=fscanf([id], 'formato',[tam])</code>	
var =array donde se meten los datos leídos lee = cantidad de datos leídos tam (si no se indica no se posiciona el puntero):	
<ul style="list-style-type: none"> <i>n</i>: lee hasta el número de elementos indicado <i>inf</i>: lee hasta el final <i>[n,m]</i>: va metiendo en la matriz indicada hasta completar el tamaño nxm. 	

Guardar variables en un archivo	
<code>save [<arch>][<v1> <v2> ...][-ascii][-tabs]</code>	
Si no se indica <arch> se guarda en matlab.mat	
-ascii guarda los datos en texto plano	
-tabs para indicar que utilice tabuladores como separador en texto plano	

Leer variables de un archivo	
<code>load [<arch>]</code>	