

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**  
**FACULTAD DE CIENCIAS QUÍMICAS FÍSICAS Y MATEMÁTICAS**  
**DEPARTAMENTO ACADÉMICO DE MATEMÁTICAS Y ESTADÍSTICA**



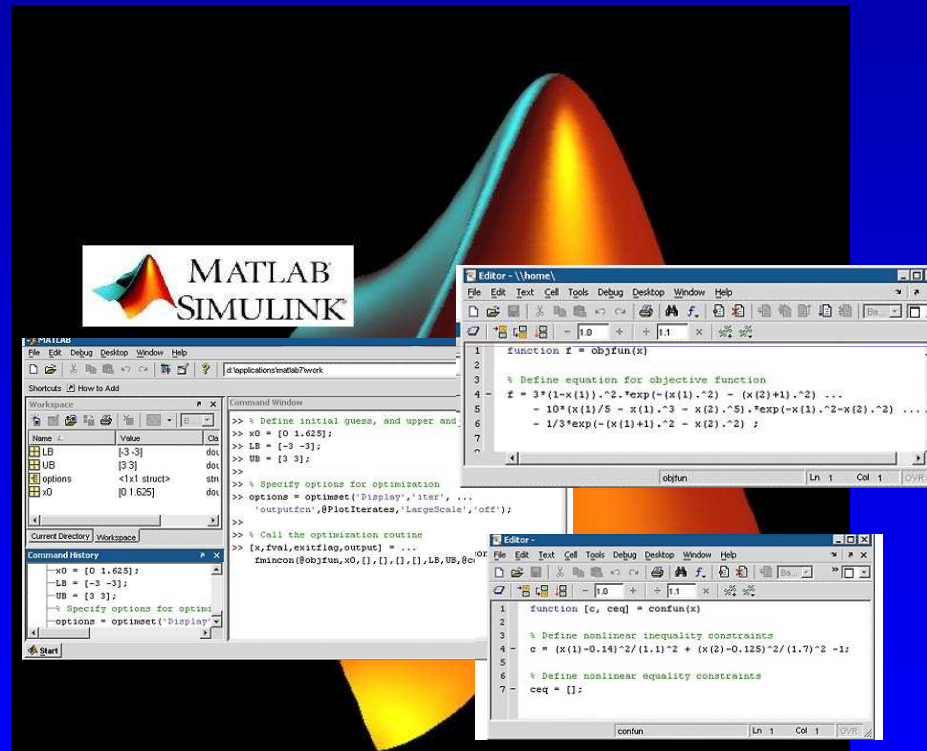
**UNSAAC**

**Lic. Guillermo Mario Chuquipoma Pacheco**


**[mariochuqui@hotmail.com](mailto:mariochuqui@hotmail.com)**

**[www.mariochuqui.jimdo.com](http://www.mariochuqui.jimdo.com)**

# M-files



# Temario General

1. Introducción.
2. Interfaz de usuario de MATLAB.
3. Variables del MATLAB.
4. Ploteo y visualización de datos.
5. **m-Files.** 
6. Estadística básica y análisis de datos.
7. Tipos de datos.
8. Entrada y salida de datos.
9. Programación.
10. Construcción de interfaces de usuarios gráficas.

# Objetivos

- Los m-files permiten codificar programas en MATLAB.
- Esta sección da a una visión general de cómo escribir, editar, ejecutar y depuran m-files.
- La distinción entre un script y un m-file es realizada.
- Se presenta elementos de estructuras básicas de programación.
- Desarrollar ejercicios de programación.

# Archivos m-file

- Contienen órdenes de MATLAB
- Se invocan desde la ventana de órdenes, o desde otro archivo.m
- Se editan y graban como ficheros ASCII.

# Funciones m-file

- Empiezan por:

```
function y = nomdefun(x)
```

- Las variables definidas no modifican las existentes en el espacio de trabajo.
- Extienden las funciones de MATLAB.
- Permiten el paso de parámetros.

# Funciones m-file

```
function r = ourrank(X,tol)
% rank of a matrix
s = svd(X);
if ( nargin == 1 )
    tol = max(size(X)) * s(1)* eps;
end
r = sum(s > tol);
```

**Argumentos de entrada**  
**Multiple**  
**usar ( )**

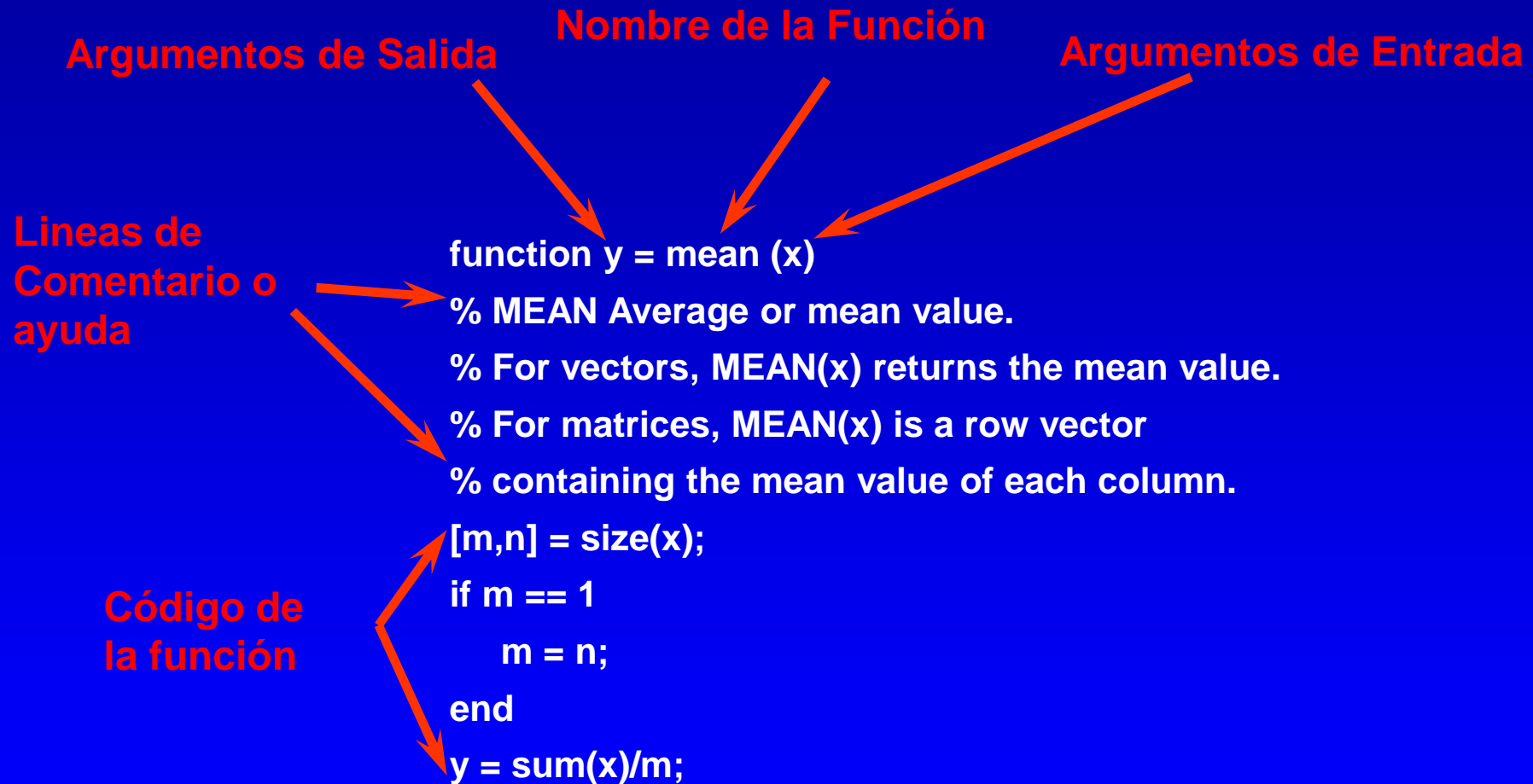
» `r=ourrank(rand(5),.1);`

**Argumentos de Salida**  
**multiple, usar [ ]**

» `[m std]=ourstat(1:9);`

```
function [mean,stdev] = ourstat(x)
[m,n] = size(x);
if m == 1
    m = n;
end
mean = sum(x)/m;
stdev = sqrt(sum(x.^2)/m - mean.^2);
```

# Partes básicas de una función m-file





# Archivos de script y de funciones

- **Archivo de Script**

- Trabaja como comandados ingresados en el command windows
- Las variables se almacenan en el área de trabajo

- **Archivo de Función:**

- Permite al programador crear sus propias funciones.
- Todas las variables en la función son locales
- Toda la información debe ser pasada a la función como parámetros.
- Las Subfunctions son soportadas.
- Permite la recursividad

# Tipos de variables en matlab

- **Variables locales**

- Una variable en MATLAB tiene ámbito local, existe solo en el espacio de trabajo
- La variable desaparece cuando el área de trabajo deja de existir
- Cada función define su propia área de trabajo, una variable definida en una función es local a esa función.
- Las variables definidas fuera la función deben ser pasadas como argumentos.
- Todos los argumentos son pasados por valor.

# Tipos de variables en matlab

- **Variables globales.**

- Están disponibles múltiples áreas de trabajo.
- Tienen que declararse explícitamente como globales
- Es una mala práctica de programación.

```
function h = falling(t)
global GRAVITY
h = 1/2*GRAVITY*t.^2;
```

# La instrucción WHILE

- Bucle controlado por una condición.
- Sintaxis:

```
while condición  
    instrucciones  
end
```

- Las instrucciones se repiten mientras la condición se verifique.

# La instrucción FOR

- Bucle que se repite un número de veces.
- Sintaxis:

```
for x = array  
    instrucciones  
end
```

- Las instrucciones se ejecutan una vez para cada columna en el array.
- Podemos anidar sentencias for.

# La instrucción IF

- Bifurcación condicional.
- Sintaxis:

```
if condición  
    instrucciones  
end
```

- Las instrucciones se realizan si la condición se verifica.

# La instrucción SWITCH

```
method = 'Bilinear';  
switch lower(method)  
    case {'linear','bilinear'}  
        disp('Method is linear')  
    case 'cubic'  
        disp('Method is cubic')  
    otherwise  
        disp('Unknown method.')
```

end

# Operadores lógicos y relacionales

## Operadores lógicos:

Conjunción	<b>&amp;</b>
Disyunción	<b> </b>
O exclusiva	<b>xor</b>
Negación	<b>~</b>

## Operadores de comparación:

Mayor, menor	<b>&gt;, &lt;</b>
Mayor o igual	<b>&gt;=</b>
Menor o igual	<b>&lt;=</b>
Igual	<b>==</b>
Distinto	<b>!=</b>

**Nota:** ~ es [Alt] + 126



# Ejercicio 1

- Desarrollar los siguientes programas en matlab:
  1. Desarrollar una función para calcular la distancia que existe entre un par de puntos en un espacio 3-D. Amplíe su función para que trabaje con más de un par de puntos.
  2. Calcular todos los pares de números en un arreglo tal que sumen cero.
  3. Solucionar un sistema de  $N$  ecuaciones lineales y  $N$  variables, no use la función `linsolve`.
  4. Generar los primeros  $k$  números de la serie de fibonacci.
  5. Crear un arreglo de tamaño  $N$  tal que todos sus elementos sean diferentes pero se ubiquen en posiciones aleatorias.

# Ejercicio 2

- Desarrollar los siguientes programas en matlab:
  1. Asuma que usted recibe una matriz de 2 columnas con los resultados de un experimento, la primera es el valor calculado y la segunda el valor deseado. Calcule el grado de éxito del experimento.
  2. Desarrollar la función select (tabla, índice, columnas, valor), donde:
    - tabla es una matriz que contiene los datos.
    - índice es el número de columna donde se busca la coincidencia con valor.
    - valor es el valor a buscar.
    - columnas, con las columnas a recuperar

***GRACIAS POR SU ATENCIÓN***

**Lic. Guillermo Mario Chuquipoma Pacheco**

**mariochuqui@hotmail.com**

**<http://www.mariochuqui.jimdo.com>**



**UNSAAC**