

DISEÑE SU INTERFAZ GRÁFICA CON MATLAB

*Miguel Ramón Ortegón Gutiérrez
Ingeniero Mecánico, M.Sc I.S
Profesor Asociado-Departamento de Ingeniería de Sistemas
Universidad Nacional de Colombia, Bogotá*

Resumen

Las aplicaciones de carácter científico o tecnológico cuentan, para su desarrollo, con el apoyo de una poderosa herramienta de cálculo numérico, que permite abordar problemas que demanden variada y compleja manipulación matemática de manera eficiente y con ayudas gráficas. Su diseño funcional le permite usar con facilidad toda la gama de comandos en forma interactiva, o en modo programas que han sido implementados como parte del lenguaje, o los nuevos comandos implementados por el usuario. Problemas tradicionales, relativos al manejo de matrices o álgebra lineal, cálculo diferencial e integral, ecuaciones diferenciales, raíces de funciones, solución de sistemas de ecuaciones, etc., hasta algunos más especializados como el procesamiento digital de señales, redes neuronales, filtros, pueden ser tratados con este lenguaje. Con el presente artículo se desea difundir el uso de interfaces gráficas que provee Matlab, para hacer más

modernas, presentables y manejables aplicaciones de ingeniería que con otras herramientas podrían resultar más complicadas y engorrosas.

INTRODUCCIÓN

Matlab es un lenguaje de tipo funcional, especialmente desarrollado para apoyar la gestión científico-tecnológica en forma eficiente, amigable y sencilla. Su estructura provee de un conjunto de poderosas funciones numéricas y gráficas, que cubren un amplio espectro de las exigencias matemáticas y de imagen de una aplicación en el campo de la ingeniería. Complementaria e integralmente a su funcionalidad, Matlab incluye un lenguaje de tipo procedimental, mediante el cual pueden estructurarse, nuevas unidades lógicas funcionales, que pueden agregarse a las ya existentes o ser usado para controlar el flujo de ejecución de un programa.

El presente artículo pretende mostrar únicamente el manejo de los comandos que facilitan la construcción de una interfaz gráfica para la interacción hombre-máquina, con la intención de proporcionar una herramienta adicional de trabajo, que mejora la calidad del *software* desarrollado.

I. INTERFAZ GRÁFICA

La interfaz gráfica es el espacio de diálogo entre usuario y computador, con el objeto de intercambiar información en condiciones de alta eficiencia, facilidad de uso y calidad estética.

El dispositivo utilizado para tal fin es la pantalla del computador, sobre la cual se despliegan los diferentes objetos gráficos que controlan el flujo de operaciones.

II. OBJETOS GRÁFICOS

Constituyen los componentes básicos de la interfaz, los cuales tienen asociados parámetros y funciones que definen su comportamiento y a partir de los cuales se construye la estructura de comunicación entre usuario y aplicación.

A. Ventana

Región rectangular de la pantalla que incluye los elementos de la interfaz. Las propiedades definibles desde Matlab son las siguientes:

- Ubicación de la ventana en el área de pantalla.
- Ancho y alto de ventana.
- Etiqueta de ventana.
- Color fondo de ventana

1. Sintaxis Matlab

```
manejo=figure('parametro 1','valor 1',.....'parametro n','valor n');
```

Los argumentos del comando **figure** son parejas de parámetro con su respectivo valor. La variable de asignación (manejo) se usa para identificar la ventana.

Los parámetros pueden ser:

Color: usado para definir el color del fondo de ventana. Se define en un vector con un valor entre 0 y 1 para indicar la intensidad del rojo, verde y azul, los cuales combinados producen la gama de colores.

Numbertitle: usado para indicar si debe aparecer o no el número de figura que coloca Matlab.

Menubar: para aceptar o no la barra estándar de menús, colocada por Matlab.

Name: para asignar el título de la ventana.

Position: para indicar la ubicación de ventana. Se define como un vector de 4 elementos, separados por comas o blancos así: [Abscisa inferior ordenada inferior deltax deltay].

Units: indica el tipo de unidad usado para dimensionar la pantalla.

2. Ejemplo 1

Dibujar una ventana cuyo vértice inferior izquierdo se encuentre en las coordenadas (170,190), con un ancho de 300 y altura de 100. Debe eliminarse la barra estándar de menús al igual que el número indicador de ventana y el título para ésta debe ser 'Universidad Nacional de Colombia'. El color de fondo, rojo.

Desarrollo

```
f1=figure('units','points','color',[1.0 0.0 0.0],...  
'position',[170 190 300 100],'name','Universidad Nacional de Colombia',...  
'tag','figura','menubar','none','numbertitle','off');
```

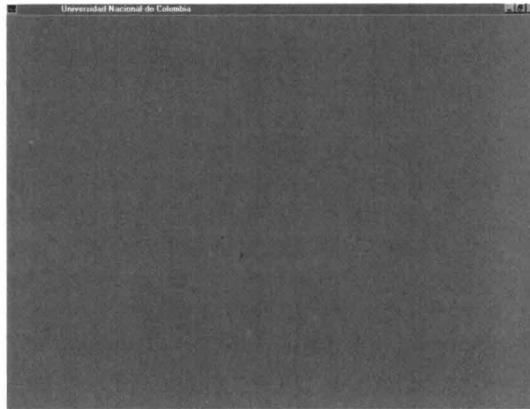


Figura 1 . Ventana general.

B. Menú deslizable

Lista de opciones ubicada generalmente en el borde superior del cuadro de interfaz. La opción deseada se activa haciendo clic con el ratón en el cajón de la opción correspondiente.

1. Sintaxis Matlab

En Matlab se especifican con el comando `uimenu`, cuya sintaxis es la siguiente:

```
menu=uimenu(gcf,'label','Cadena de caracteres del titulo',...  
'position',[x,y,deltax,deltay],...  
'backgroundcolor',[r,g,b],'callback','nombre de la funcion invocada');
```

- uimenu:** comando para creación menú interfaz de usuario.
- gcf:** ventana actual.
- label:** título del cuadro de menú.
- callback:** función invocada cuando se hace clic con el ratón.

2. Ejemplo 2

Diseñar menú de usuario 'Teoría' en la ventana corriente.

Desarrollo

```
menu1=uimenu(gcf,'label',&Teoría')
```



Figura 2. Ventana general con menú interfaz 'Teoría'.

2. Ejemplo 3

Agregar a la barra creada en el ejemplo 2, una lista de opciones cuyos títulos y funciones invocadas cuando se haga clic en el cuadro correspondiente, sean:

Títulos	Función invocada
Gauss	<i>gauss</i>
Seidel	<i>seidel</i>
Factorización	<i>lu</i>
Retornar	<i>retorno</i>

Desarrollo

```
f1=figure('units','points','color',[1.0 0.0 0.0],...  
    'position',[170 190 300 100],...  
    'tag','figura','menubar','none','numbertitle','off',...  
    'name',' Universidad Nacional de Colombia');  
menu1=uimenu(f1,'label','&teoría');  
menu11=uimenu(menu1,'label','Gauss','callback','gauss');  
menu12=uimenu(menu1,'label','Seidel','callback','seidel');  
menu13=uimenu(menu1,'label','Factorizacion','callback','lu');  
menu14=uimenu(menu1,'label','Retornar','callback','retorno');
```

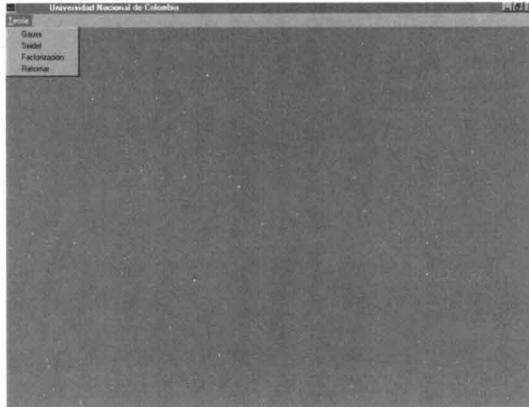


Figura 3. Ventana con interfaz de usuario y lista de opciones.

C. Botón de pulsar

Objeto de control cuya función es ejecutar la misma acción cada vez que se pulse.

Las propiedades asociadas para este objeto son: posición, tamaño, cadena de despliegue, función que se ejecuta cuando se hace clic sobre éste.

1. Sintaxis Matlab

```
Boton=icontrol('position',[x,y,deltax,deltay], 'style','pushbutton', 'string','despliegue',...
'callback','funcion a llamar');
```

icontrol es el comando de control de interfaz de usuario y *style* debe ser *pushbutton*.

2. Ejemplo 4

Colocar un botón en la mitad inferior de la ventana , que despliegue la palabra 'Continue' y active la función 'seguir' , cuando se haga clic sobre él.

Desarrollo

```
f1=figure('unit','points','color',[1.0 0.0 0.0],...
'position',[170 190 300 100],...
'tag','figura','menubar','none','numbertitle','off',...
'name',' Universidad Nacional de Colombia');
botonp=icontrol(gcf,'style','pushbutton', 'position',[150 30 50 25],...
'string','Continue', 'callback','seguir');
```

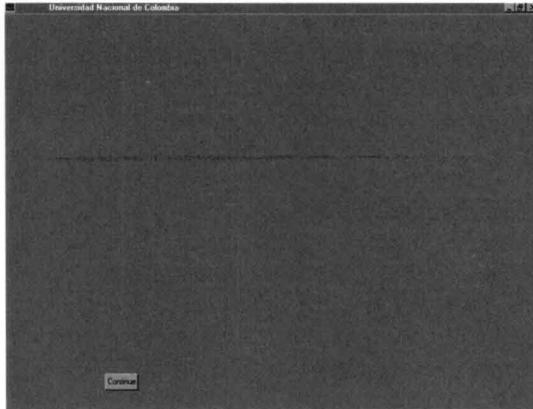


Figura 4. Ventana con botón de pulsar.

D. Casilla de verificación

Objeto rectangular que se usa especialmente en aquellas circunstancias en las cuales se desea realizar operaciones de selección simple, activando o apagando para ello un *switch*.

1. Sintaxis Matlab

Manejo=**uicontrol**(*gcf*, 'style', 'checkbox', 'value', entero,...
'position',[x y deltax deltay], 'string', 'on/off', 'callback', 'funcion matlab');

entero: valor con el cual se inicia el parámetro 'value'.

On/off: palabras para desplegar en la caja.

Instrucciones: acciones en sintaxis de Matlab de acuerdo con cada pulsación.

2. Ejemplo 5.

Colocar una casilla de verificación, la cual elimine o dibuje los ejes para un gráfico.

Desarrollo

```
f1=figure('unit','points','color',[1.0 0.0 0.0],'position',[170 190 300 100],...
'tag','figura','menubar','none','numbertitle','off',...
'name',' Universidad Nacional de Colombia');
x=-1:0.2:1;
plot(x,exp(x)-x); % graficar la función exponencial
xlabel('X'); % etiqueta para el eje X
ylabel('EXP(X)-X'); % etiqueta para el eje Y
```

```

Casilla=icontrol(gcf,'style','checkbox','value',1,...
'position',[150 30 65 25],'string','on/off',...
'callback',...
[...
'if get(casilla,"value")==0, axis off;',...      % los ejes se trazan o no cada vez que se
'elseif get(casilla,"value")==1, axis on,end'...  % pulse en la casilla (on/off)
]);

```

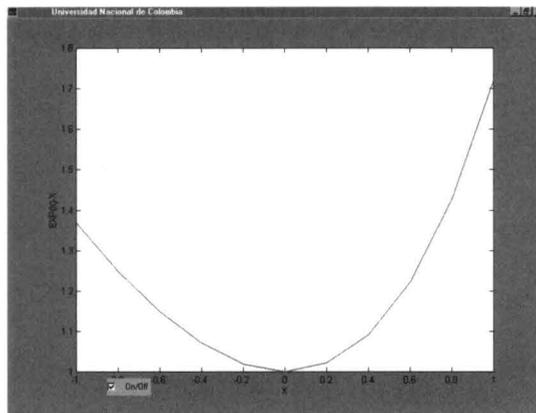


Figura 5. Ventana con casilla de verificación (on).

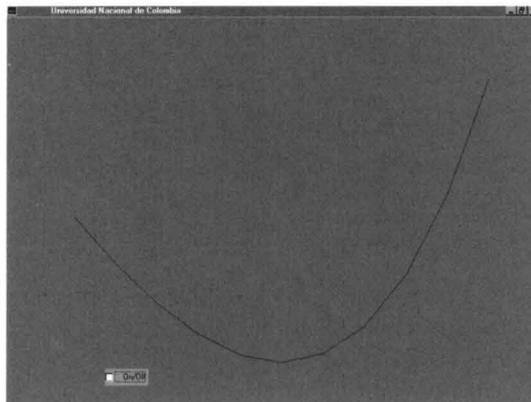


Figura 6. Ventana con casilla de verificación (off).

E. Botón de radio

Objeto circular que precede el nombre de una opción. Los botones de radio para una lista de opciones son mutuamente excluyentes. Solamente se activa la opción sobre la cual se hace clic.

1. Sintaxis Matlab

```
Botonp=uicontrol('style','radiobutton','position',[x y deltax deltay],...  
'string','cadena a desplegar en el botón','callback','función invocada');
```

radiobutton: parámetro del control de interfaz, para indicar botón de radio.
[x y deltax deltay]: posición y tamaño del botón.
Cadena a desplegar...: cadena de caracteres para desplegar sobre el botón.

2. Ejemplo 6

Colocar 2 botones de radio que desplieguen las cadenas 'Boton 1' y 'Boton 2'.

Desarrollo

```
f1=figure('units','points','color',[1.0 0.0 0.0],...  
'position',[170 190 300 100],...  
'tag','figura','menubar','none','numbertitle','off',...  
'name','Universidad Nacional de Colombia');  
f1=figure('units','points','color',[1.0 0.0 0.0],'position',[170 190 300 100],...  
'tag','figura','menubar','none','numbertitle','off',...  
'name','Universidad Nacional de Colombia');  
botonp1=uicontrol(gcf,'style','radiobutton','position',[150 50 60 25],...  
'string','Boton 1','callback','radio1');  
botonp2=uicontrol(gcf,'style','radiobutton','position',[150 30 60 25],...  
'string','Boton 2','callback','radio2');
```

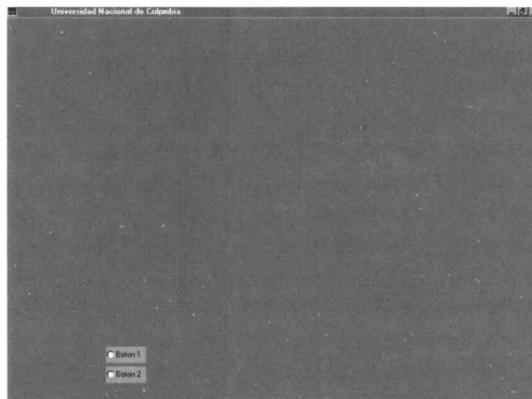


Figura 7 Ventana con dos botones de radio.

F. Control deslizante

Objeto que realiza la función de un dispositivo analógico, con el fin de permitir la modificación de un parámetro.

1. Sintaxis Matlab

```
Manejo=uicontrol(gcf,'style','slider','min',minimo,'max',maximo,...  
    'position',[x y deltax deltay,'value',variable,'callback',...  
    'variable=get(manejo,"value"); set(variable de manejo parámetro a modificar,  
    "color",[rojo gris azul])');
```

- style** : parámetro de control de interfaz. Toma el valor 'slider'.
- min** : límite inferior del intervalo de variación del parámetro a modificar.
- max** : límite superior del intervalo de variación del parámetro a modificar.
- value** : nombre del parámetro que ha de modificarse.

2. Ejemplo 7

Colocar un control deslizante para modificar el color rojo de manera progresiva, desde su nivel básico hasta el más intenso.

Desarrollo

```
clf; clear;  
h6=figure(1);  
R=0; g=0; B=0;  
Set(h6,'position',[60 200 380 280],...  
    'name','Universidad Nacional : Demo RGB','numbertitle','off',...  
    'color',[0 0 0]);  
Ah=axes('position',[0.4 0.1 0.55 0.50]); % posición y tamaño de la caja de color rojo.  
'box','on',...  
'color',[R G B],...  
'xcolor',[0 0 0],'ycolor',[0 0 0];  
deslizar=uicontrol(gcf,'style','slider',... % se define control deslizante  
    'min',0,'max',1,...  
    'position',[150 30 45 25],...  
    'value',R,...  
    'callback',  
    'R=get(deslizar,"value"); set(ah,"color",[R G B])');  
Ah es la variable de manejo del parámetro a modificar..
```

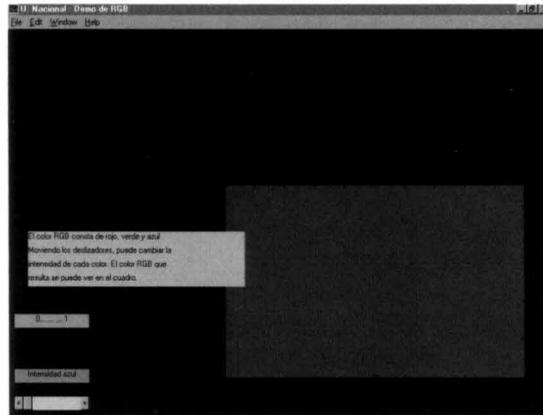


Figura 8 Ventana con control deslizante.

G. Texto editable

Objeto que permite el manejo de cadenas de caracteres.

1. Sintaxis Matlab

```
Manejo=icontrol('style','edit',...
    'position',[x y deltax deltay],...
    'string','cadena',...
    'callback','funcion invocada');
```

style : parámetro de control de interfaz. Debe ser 'text'.

Position : posición donde se desea colocar el texto.

String : cadena a desplegar.

2. Ejemplo 8

Desplegar una cadena con la palabra 'Digite su clave' , abriendo a su lado una caja para colocar la respuesta.

Desarrollo

```
f1=figure('units','points','color',[1.0 0.0 0.0],...
    'position',[170 190 300 100],...
    'tag','figura','menubar','none','numbertitle','off',...
    'name',' Universidad Nacional de Colombia');
manejo1=icontrol('style','text','position',[10 20 90 20],... % texto de despliegue
    'string','Digite su clave');
manejo2=icontrol('style','edit','position',[110 20 40 20],... % caja para respuesta
    'callback','clave=get(manejo2,"string")'); % guarda respuesta en 'clave'.
```

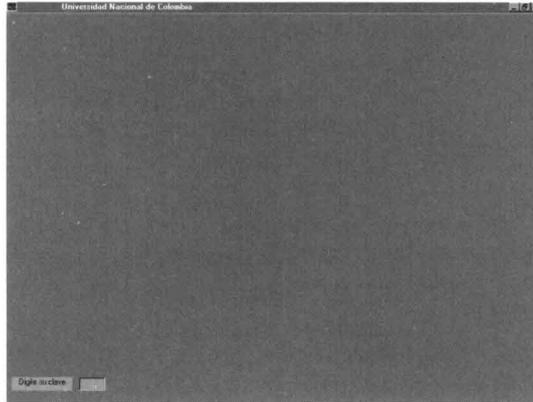


Figura 9. Ventana con edición de texto.

H. Menú desplegable

Objeto que presenta una lista de opciones en cualquier lugar de la pantalla. Haciendo clic se selecciona la opción deseada.

1. Sintaxis Matlab

```
Manejo=icontrol('style','popup',...
    'string','opcion1 | opcion2 | opcion3.....opcion n',...
    'position',[x y deltax deltay],...
    'callback',...
    [...
    instrucciones de manejo
    ]);
```

style : palabra de control de interfaz de usuario. Debe ser 'popup'.
String : se colocan separadas por la barra vertical, las diferentes opciones.
Instrucciones de... : instrucciones Matlab para interpretar la opción seleccionada.
Position : posición y tamaño del menú desplegable.

2. Ejemplo 9

Colocar un menú desplegable que presente las opciones y acciones indicados a continuación:

Opción	Acción
Teoría	Ejecutar función teoría.
Cálculos	Ejecutar función cálculos.
Gráficas	Ejecutar función gráficos.

Desarrollo

```
Manejo=icontrol('style','popup','position',[10 60 90 20],...  
'string','Teoria | Calculos | Graficas',...  
'callback',...  
[...  
'x=get(manejo,"value");',...  
'if x==1, teoria,',...  
'elseif x==2, calculos,',...  
'elseif x==3, graficos,', 'end'...  
]);
```



Figura 10. Ventana con menú desplegable.

III. APRECIACIONES

Existen en el mercado otras herramientas de *software* que permiten diseñar interfaces de usuario de muy buena calidad, independientes de Matlab y de manejo más complicado.

El diseño de interfaz con Matlab puede ser menos poderoso que otros paquetes de *software*, pero suficientemente bueno para aplicaciones de ingeniería que requieran interactuar con usuarios.

Los programas de interfaz escritos en Matlab son más compactos y más legibles, haciendo, por tanto, más fácil su mantenimiento.

El tiempo de aprendizaje es inferior al de otros lenguajes usados para el mismo fin.

Los requerimientos de *hardware* y *software* no son demasiado exigentes.

Ofrece gran cantidad de ayudas en línea para orientar su uso y facilitar la depuración de programas.

Para cambiar o conocer los atributos de las distintas entidades gráficas pueden utilizarse los comandos 'get' y 'set'.

BIBLIOGRAFÍA

1. NAKAMURA, Shoichiro, *Análisis numérico y visualización gráfica con Matlab*, Prentice Hall, 1997.
2. MATWORKS. *Building a graphical user interface*, MatWorks , 1993.
3. *Ayuda en línea (Help) de Matlab*.