Introducción a la Programación Visual con Matlab

Torres Ramírez Luis Alberto

1. Breve descripción de ambiente de desarrollo de Interfaz de Usuario

El ambiente de desarrollo de Interfaz de Usuario en Matlab (GUIDE) contiene un conjunto de herramientas para crear interfaces graficas muy parecidas a las aplicaciones Windows. Estas herramientas simplifican el proceso de creación y de programación.

Cuando abrimos un GUI con GUIDE, Matlab despliega el editor de diseño (Layout), el cual mostramos en la figura 1.

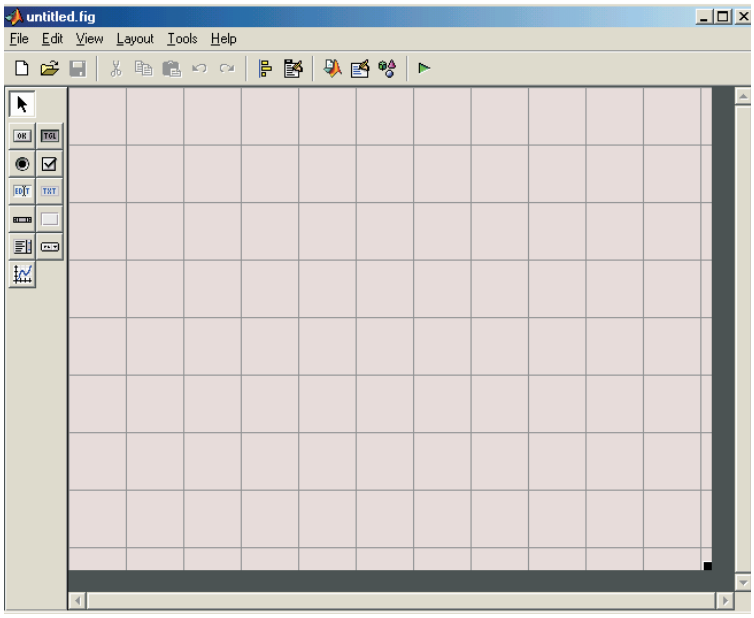

Figura 1: El editor de diseño

En el área de diseño, también conocida como formulario y que en adelante asi le llamaremos, podemos colocar los elementos que se encuentran en las herramientas como un boton de comando (push botton), menús despegables (popup menu), ejes (axes), etc., de las cuales daremos una descripción mayor adelante.

Cuando nosotros guardamos nuestro diseño grafico creado en la ventada GUIDE, Matlab crea automáticamente el archivo "m" correspondiente, el cual tendrá el mismo nombre que el creado en GUIDE, el cual tiene la extensión fig. El archivo 'm' contiene los callback correspondientes a cada componente, en la figura 2 mostramos, borrando los comentario, el archivo que se genera de un formulario vacío como el de la figura 1.

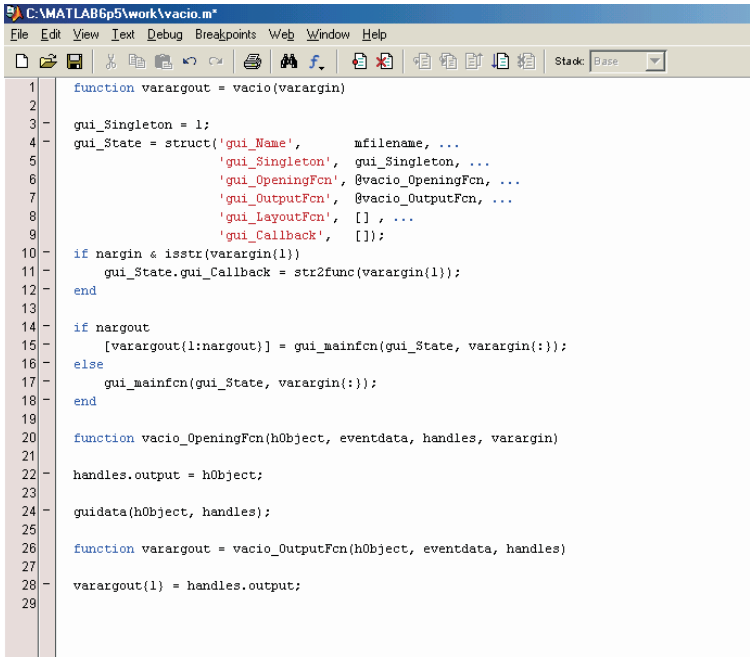

Figura 2: Archivo m de un formulario vacío

2. Plantillas

Las plantillas son ejemplo de interfaces de usuario (ver figura 3) que pueden ser modificados, los cuales son:

a) Blank GUI (Default)

La opción de interfaz gráfica de usuario en blanco (viene predeterminada), nos presenta un formulario vaciío, en el cual podemos diseñarlo a nuestro gusto.

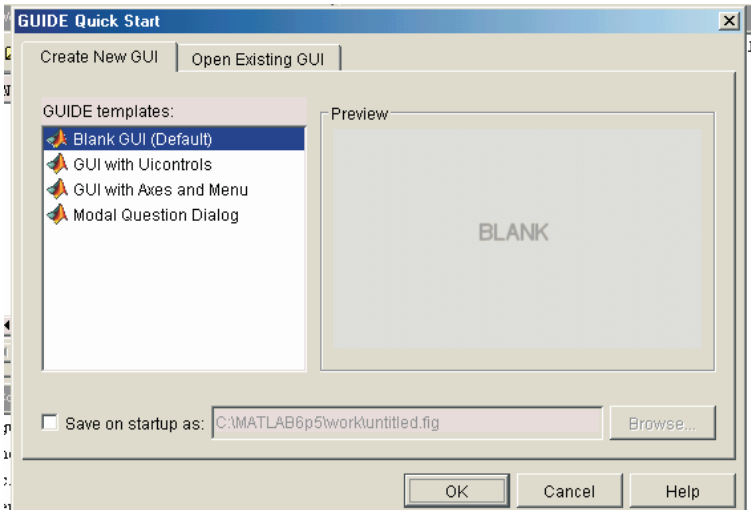

Figura 3: El editor de diseño

b) GUI with Uicontrols

Esta opción presenta un ejemplo en el cual se calcula la masa, dada la densidad y el volumen, en alguno de los dos sistemas de unidades, podemos ejecutar (correr) este ejemplo y obtener resultados.

c) GUI with Axes and Menu

Esta opción es otro ejemplo el cual contiene el menu File con las opciones Open, Print y Close, en el formulario tiene un Popup menu, un push botton y un objeto Axes, podemos correr el programa seleccionando alguna de las seis opción que se encuentran en el menu despegable y haciendo clic en el boton de comando (update).

d) Modal Question Dialog

Con esta opción se muestra en la pantalla un cuadro de diálogo comun, el cual consta de una pequeña imagen, una etiqueta y dos botones Yes y No, dependiendo del boton que se presione, el GUI retorna el texto seleccionado (es decir, la cadena de caracteres 'Yes' o 'No').

3. Cuadro de Herramientas

El cuadro de herramientas contiene los controles que se pueden utilizar en nuestra interfaz grafica de usuario. Estos componentes con objetos uicontrols de Matlab y son programables a través de los callback (las funciones que se ejecutan cuando el usuario acriva algun componente en la GUI). Los componentes del cuadro de herramientas que son de nuestro interés son:

1. Push Buttons

   Los botones de comando generan una acción cuando son presionados. Cuando se hace clic en un boton de comando, da la apariencia de ser presionado, cuando dejamos de hacer clic tiene apariencia normal (no presionado). En el editor de propiedades (figura 4) podemos cambiar la apariencia y texto predefinido, nosotron solo nos interesaremos en el tamaño y el texto que tiene por default, si tiene tiempo puede hacer algunos otros cambio y revisar la salida, o bien puede consultar la ayuda.

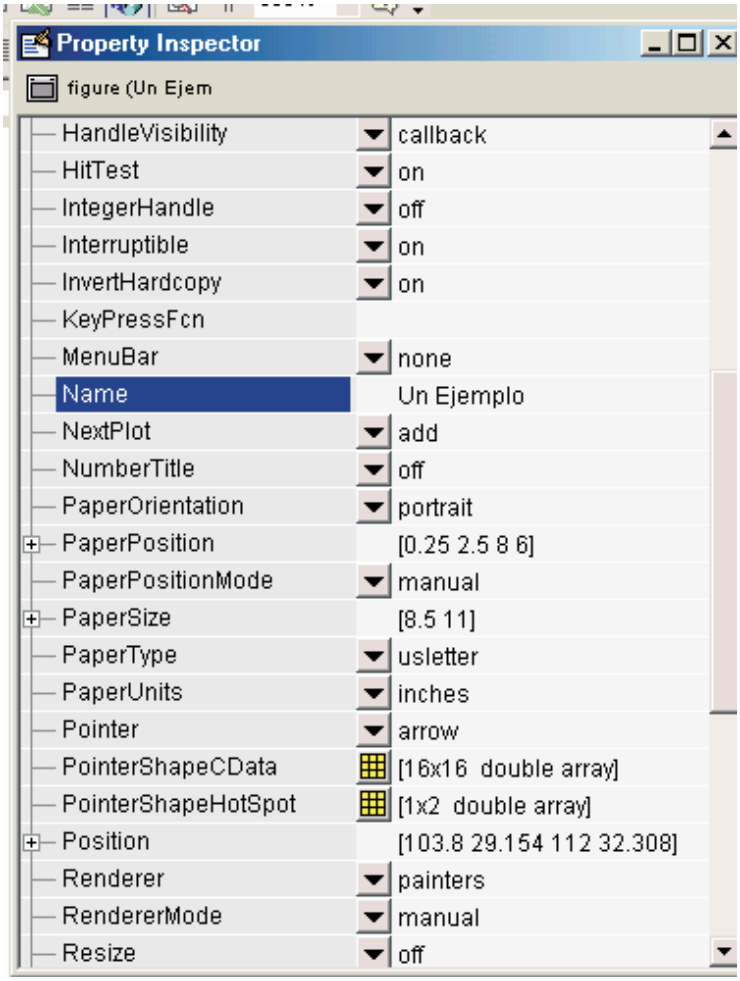

Figura 4: El editor de propiedades

En un botón de comando, la propiedad 'String' son los caracteres que

nosotros deseamos que se muestren en el botón. mientras que la propiedad 'Tag' es el nombre que le pondremos a la subfunción callback en archivo m, es decir es el nombre del comando, nosotros no modificaremos esta propiedad, para evitarnos confusiones con los nombres. Su programación de callback se basa en la programación normal de Matlab, dentro de la funcion correspondiente.

```
function pushbutton1_Callback(hObject, eventdata, handles)

A(1,1)=str2num(get(handles.edit1,'String'));
A(1,2)=str2num(get(handles.edit2,'String'));
A(2,1)=str2num(get(handles.edit4,'String'));
A(2,2)=str2num(get(handles.edit5,'String'));
```

En este pequeño ejemplo estamos obteniendo (función get) el valor que se tiene en la propiedad string, de unos componentes edit, la instrucción handles se debe a que cada componente es un objeto, los cuales perteneces a una estructura, no vamos a profundizar mas en esto, pero es necesario que lo tengan en cuenta ya que en el desarrollo de este curso se usarán estas instrucciones. Otra intrucción que aparece es la instrucción str2num, la cual convierta a número una cadena de caracteres, para más detalles vea los manuales de los cursos basicos de Matlab.

2. Radio buttons

   Radio button es similar a check boxes, con la diferencia que los radio buttons se tienen que excluir mutuamente de entre un grupo de estos botones. Para activar un radio button se hace clic sobre la opcion que deseamos seleccionar. Estos botones tienen dos estados, seleccionado y no seleccionado y se observa en la propiedad Value, si esta en Max, el boton esta seleccionado, si su valor es cero, no esta seleccionado. En el siguiente texto, estamos suponiendo que tenemos cuatro radio buttons y vamos a excluirlos

```
function radiobutton1_Callback(hObject, eventdata, handles)
off=[handles.radiobutton2, handles.radiobutton3, handles.radiobutton4];
set(off,'Value',0)
%contunua el callback
```

   posteriormente se hace algo similar para cuando se seleccione alguno de los otros radio buttons, cambiando la numeración de los botones.

3. Check boxes

   Esta opción es similar a la anterior, pero no necesitamos excluir mutuamente los objetos check boxes utilizados.

4. Edit Text

   Estos controles son campos en los cuales los usuarios pueden modificar las cadenas de texto. Use el editor de texto cuando desea que ingresen texto o datos haciendo uso de las instrucciones aprendidas en los cursos de Matlab. Una última observación, la propiedad string cambia por la propiedad text.

5. Static text

   El texto estático mejor conocido como etiquetas, despliegan lineas de texto. estas no pueden ser modificadas por el usuario.

6. Slider

   Barra de desplazamiento, esta opción acepta como entrada datos numéricos con un rango específico, para mover la barra de desplazamiento, ésta se mueve presionando con el mause sobre la barra.

7. Frames

   Este componente en realidad son cajas que contienen o delimitan regiones de una figura (ventana), estas cajas suelesn ser introducidas para seperar por grupos varios controles para mayor facilidad.

8. List boxes

   Despliega una lista y permite al usuario seleccionar algun elemento de la lista.

9. Pop-Up Menus

   Despliega de opciones cuando hacemos clic en la flecha que tiene a un lado. Para agregar elementos a la lista, seleccionamos el boton que se encuentra a un lado de la opcion string del cuedro de propiedades e introducimos uno a uno cada elemento; dicho botón esta entre string y lo que se pondrá. Para programar este elemento, es necesario tener presente las bifurcaciones o condicionales, en especial la instrucción switch.

10. Axes

    La mejor utilidad comparando Matlab con otros programas visuales, en los ejes podemos desplegar gráficas o imágenes, podemos controlar los objetos graficos y sus características. Axes no es un uicontrol, pero puede ser programado al momento de ejecutar su callback cuando los usuario hagan clic sobre los ejes. Cuando sean necesaios varios ejes, debemos seleccionar explícitamente cual de los ejes que tenemos será utilizado para graficar, para esto usamos el comando axes, por ejemplo:

    axes(handles.axes1)

    y luego lo que se hará. Aquí podemos también decidir cuando se mostrará una grafica y cuando no, para esto hacemos uso de la propiedad visible, la cual tiene las opciones on u off.

4. Menus

Cear un menú en Matlab, es sencillo, y para ello basta con hacer uso del editor de menus (figura 5)

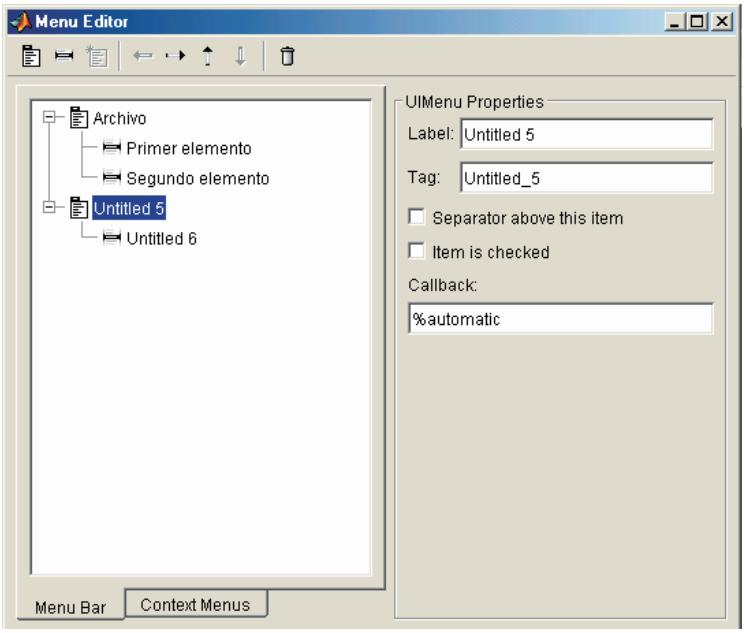

Figura 5: El editor de menus

En su barra de herramientas tenemos varias opciones, las opciones que nos interesan son nuevo menu, nuevo elemento y eliminar, el cual es el último. Debemos cambiarle la etiqueta que aparece y podemos cambiarle el nombre. La instrucción callback es para hacer llamada a su correspondiente callback (aunque podemos realizar directamente algunas instrucciones).

5. Ejemplos

Con lo que hemos revisado, ya estamos en la posibilidad de realizar algunos ejemplos sencillos, debido al tiempo que se tiene para este curso crearemos solo algunos, haciendo uso de la mayoría de los objetos vistos y, esperamos que los interesados puedan seguir trabajando para tener un mayor manejo en este entorno de trabajo.

Para comenzar, escribimos la instrucción 'guide' en el promt de Matlab y presionamos enter, aparecerá un cuadro de dialogo como el de la figura 2, seleccionamos la opcion de dialogo y hacemos clic en aceptar, una vez que a aparecido

el formulario lo guardamos con el nombre de modaldlg; ahora hacemos clic en nuevo y seleccionamos una en blanco y hacemos clic en aceptar. En este ejemplo solo haremos uso del cuadro de diálogo, así pues, ponemos una etiqueta y un botón de comando, debe quedar como la figura 6.

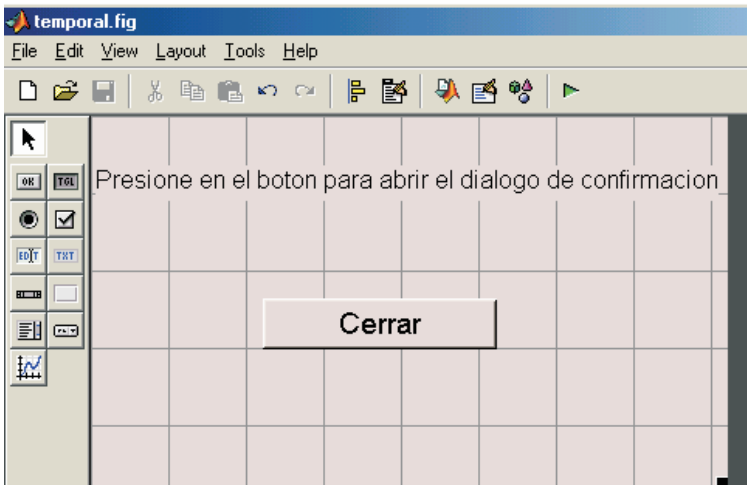

Figura 6: Primer ejemplo

una vez que guardamos nuestro archivo aparece el editor de Matlab, agregamos las linas correspondientes

```
function varargout = temporal(varargin)
% Aqui ponemos lo que deseamos que aparezca en la ayuda

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @temporal_OpeningFcn, ...
                   'gui_OutputFcn',  @temporal_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
```

```
end

function temporal_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = temporal_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

%-------------------------------------------------------------
function pushbutton1_Callback(hObject, eventdata, handles)

%despues de los comentarios agregamos
pos_size=get(handles.figure1,'Position');
user_response=modaldlg('Title','Confirmar para cerrar');
switch user_response
case{'No'}
 %no hacemos nada
case 'Yes'

delete(handles.figure1) %se cierra la ventana principal
end
```

Ahora, ponganlo en practica y vean que realiza.

Nuestro siguiente ejemplo encuentra la solución (no esta demasiado completo ya que falta la opción cuando las lineas no se intersectan) de un sistema de dos ecuaciones con dos incógnitas, en este ejemplo hacemos uso de 6 objetos de texto, 8 etiquetas, un boton y uno de ejes. El formulario terminado debe tener una apariencia como la de la figura 7

Una vez realizado esto, guardamos nuestro archivo y una vez abierto el editor de Matlab, agregamos unas lineas en el boton de comando, agregelas por favor, todo el archivo debe quedar mas o menos así:

```
%estas lineas hay que modificar
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
A(1,1)=str2num(get(handles.edit1,'String'));
A(1,2)=str2num(get(handles.edit2,'String'));
A(2,1)=str2num(get(handles.edit4,'String'));
A(2,2)=str2num(get(handles.edit5,'String'));

b(1)=str2num(get(handles.edit3,'String'));
```

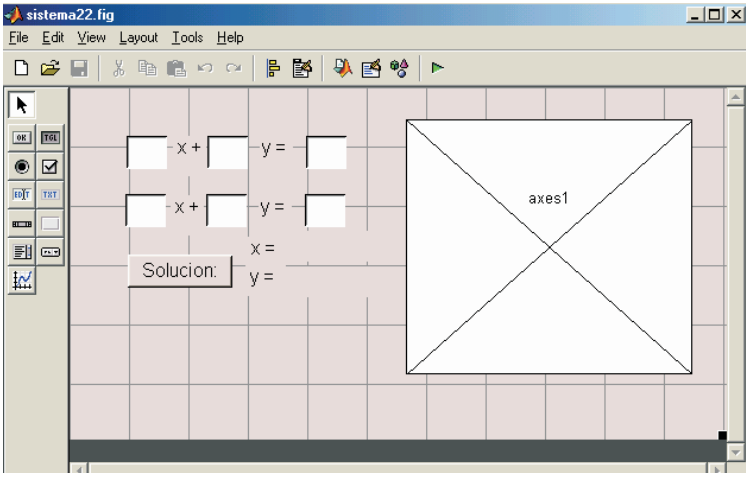

Figura 7: Otro ejemplo

```
b(2)=str2num(get(handles.edit6,'String'));

x=A\(b');
set(handles.text7,'String',x(1));
set(handles.text8,'String',x(2));

vect1=x(1)-10:0.1:x(1)+10;
vect2=x(2)-10:0.1:x(2)+10;
plot(vect1,(b(1)-A(1,1)*vect1)/A(1,2),vect2,(b(2)-A(2,1)*vect2)/A(2,2));
```

Corremos nuestro programa y debe darno la siguiente ventana, representada en la figura 8

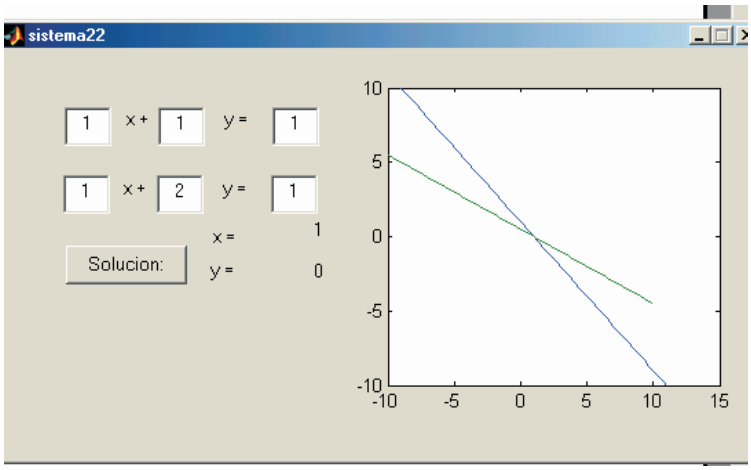

Figura 8: Solucion de un sistema de 2×2