

---

## Apéndice A

### Código Módulo Principal

```
function varargout = TutorMatlabMain(varargin)
% TUTORMATLABMAIN Application M-file for TutorMatlabMain.fig
%   FIG = TUTORMATLABMAIN launch TutorMatlabMain GUI.
%   TUTORMATLABMAIN('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 24-Mar-2003 20:03:13

if nargin == 0 % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);

    if nargin > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK

    try
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
    catch
        disp(lasterr);
    end
end

% -----
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
%disp('pushbutton2 Callback not implemented yet.')

    close (gcbf)
    close all

% -----
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton3.
%disp('pushbutton3 Callback not implemented yet.')

img1
%ejecuto aritmetica
clc
run tutormatlab6

% -----
function varargout = pushbutton4_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton4.
%disp('pushbutton4 Callback not implemented yet.')
%ejecuto algebra
```

---

---

```

img1
clc
run algebra6

% -----
function varargout = pushbutton5_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton5.
%disp('pushbutton5 Callback not implemented yet.')
%ejecuto calculo
img1

clc
run calculo6

% -----
function varargout = pushbutton6_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton6.
%disp('pushbutton6 Callback not implemented yet.')
img1

clc
run ediferencial6

% -----
function varargout = pushbutton7_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton7.
%disp('pushbutton7 Callback not implemented yet.')
img1

clc
run progra6

% -----
function varargout = pushbutton8_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton8.
%disp('pushbutton8 Callback not implemented yet.')

img1

run acerca

% -----

```

### **Código Módulo Aritmética**

```

function varargout = aritmo6(varargin)
% ARITMO6 Application M-file for aritmo6.fig
% FIG = ARITMO6 launch aritmo6 GUI.
% ARITMO6('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 22-Mar-2003 13:00:31

if nargin == 0 % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);

```

---

---

```

        guidata(fig, handles);
        if nargout > 0
            varargout{1} = fig;
        end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
    try
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
switchyard
    catch
        disp(lasterr);
    end
end

% -----
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton1.
%disp('pushbutton1 Callback not implemented yet.')
    close ( gcbf );

% -----
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton3.
%disp('pushbutton3 Callback not implemented yet.')
%se pone el texto

ponte = findobj ( gcbf, 'Tag' , 'text3' );

pon = 'En MatLab, es posible realizar las operaciones basicas tales como: +, -
, x, / ';

set ( ponte, 'String', pon );

%cambio estado del boton

cambio = findobj ( gcbf, 'Tag', 'inicio' );
set ( cambio , 'String' , 'Da enter' );

%ok

pause

pon2 = 'En el siguiente cuadro se realizan todas estas operaciones. Observe en
la ventana de comandos.';

set ( ponte , 'String' , pon2);

%cambio estado del boton

set ( cambio , 'String' , 'Inicio' );

%ok

% -----
function varargout = edit1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit1.
%disp('edit1 Callback not implemented yet.')

```

---

---

```

%defino variable
global a;
disp ('En la linea de comandos ponemos a = el numero a usar');
a = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit2.
%disp('edit2 Callback not implemented yet.')
global b;
%defino variable
disp ('En la linea de comandos ponemos b = el numero a sumar');
b = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = pushbutton4_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton4.
%disp('pushbutton4 Callback not implemented yet.')
%realizo calculos

global a
global b

%revisamos errores

if isempty (a) | isempty(b)

    run error6

else

    clc;

%preparo variables de salida

res_suma = findobj ( gcbf, 'Tag', 'suma' );
res_resta = findobj ( gcbf, 'Tag', 'resta' );
res_multiplica = findobj ( gcbf, 'Tag', 'multiplica' );
res_divide = findobj ( gcbf, 'Tag', 'divide' );

%hago calculos
disp ('Observe como se hacen las operaciones:');
disp ('a+b');
lasuma = a+b
disp ('a-b');
laresta = a-b
disp ('a*b');
lamultiplicacion = a*b
disp ('a/b');

if a == 0 & b == 0

    disp ('Ambos son cero, la division no se puede hacer.')
    ladivision = ' '

%despliego resusltados

set ( res_suma, 'String', lasuma );
set ( res_resta, 'String', laresta );

```

---

---

```
set ( res_multiplica, 'String', lamultiplicacion );
set ( res_divide, 'String', ladivision );
```

```
elseif b == 0
```

```
    disp ('Denominador cero. MatLab Regresa.')
    ladivision = a/b
```

```
    %despliego resultados
```

```
set ( res_suma, 'String', lasuma );
set ( res_resta, 'String', laresta );
set ( res_multiplica, 'String', lamultiplicacion );
set ( res_divide, 'String', ladivision );
```

```
else
```

```
    ladivision = a/b
    %despliego resultados
```

```
set ( res_suma, 'String', lasuma );
set ( res_resta, 'String', laresta );
set ( res_multiplica, 'String', lamultiplicacion );
set ( res_divide, 'String', ladivision );
```

```
end
```

```
end
```

```
end
```

```
%ok
```

```
% -----
```

### **Código Módulo Álgebra**

```
function varargout = matrixop6(varargin)
% MATRIXOP6 Application M-file for matrixop6.fig
%   FIG = MATRIXOP6 launch matrixop6 GUI.
%   MATRIXOP6('callback_name', ...) invoke the named callback.
```

```
% Last Modified by GUIDE v2.0 22-Mar-2003 13:25:04
```

```
if nargin == 0 % LAUNCH GUI
```

```
    fig = openfig(mfilename,'reuse');
```

```
    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);
```

```
    if nargout > 0
        varargout{1} = fig;
    end
```

```
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
```

---

---

```

        try
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
switchyard
        catch
            disp(lasterr);
        end
end

% -----
function varargout = edit7_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit7.
%disp('edit7 Callback not implemented yet.')
%define cada text edit

global a;

a = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit8_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit8.
%disp('edit8 Callback not implemented yet.')
%define cada text box
global b;

b = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit9_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit9.
%disp('edit9 Callback not implemented yet.')

%define cada text box
global c;

c = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit10_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit10.
%disp('edit10 Callback not implemented yet.')

%define cada text box
global d;

d = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit11_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit11.
%disp('edit11 Callback not implemented yet.')

%define cada text box
global e;

e = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit12_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit12.

```

---

---

```
%disp('edit12 Callback not implemented yet.')
```

```
%define cada text box
global f;

f = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit13_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit13.
%disp('edit13 Callback not implemented yet.')
```

```
%define cada text box
global g;

g = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit14_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit14.
%disp('edit14 Callback not implemented yet.')
```

```
%define cada text box
global h0;

h0 = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit15_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit15.
%disp('edit15 Callback not implemented yet.')
```

```
%define cada text box
global i;

i = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit16_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit16.
%disp('edit16 Callback not implemented yet.')
```

```
%define cada text box
global j;

j = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit17_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit17.
%disp('edit17 Callback not implemented yet.')
```

```
%define cada text box
global k;

k = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit18_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit18.
%disp('edit18 Callback not implemented yet.')
```

---

---

```

%define cada text box
global l;

l = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit19_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit19.
%disp('edit19 Callback not implemented yet.')

%define cada text box
global m;

m = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit20_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit20.
%disp('edit20 Callback not implemented yet.')

%define cada text box
global n;

n = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit21_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit21.
%disp('edit21 Callback not implemented yet.')

%define cada text box
global o;

o = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit22_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit22.
%disp('edit22 Callback not implemented yet.')

%define cada text box
global p;

p = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit23_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit23.
%disp('edit23 Callback not implemented yet.')

%define cada text box
global q;

q = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = edit24_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit24.

```

---



---

```

%disp('edit24 Callback not implemented yet.')

%define cada text box
global r;

r = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = pushbutton4_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton4.
%disp('pushbutton4 Callback not implemented yet.')
%realizo sumas
clc;
global a b c d e f g h0 i j k l m n o p q r;

if isempty(a) & isempty(b) & isempty(c) & isempty(d) & isempty(e) & isempty(f)
& isempty(g) & isempty(h0) & isempty(i) & isempty(j) & isempty(k) &
isempty(l) & isempty(m) & isempty(n) & isempty(o) & isempty(p) &
isempty(q) & isempty(r)

    run error6

else

disp ('Para definir una matriz solo hay que poner en la ');
disp ('linea de comandos: MatrixA = [(r1,c1),(r1,c2)];');
disp ('                [(r2,c1),(r2,c2)];');
disp ('o se puede usar el mismo renglon. ')
disp (' MatrixA = [(r1,c1),(r1,c2);(r2,c1),(r2,c2)]')

Matrix_a = [a,b,c;d,e,f;g,h0,i]
Matrix_b = [j,k,l;m,n,o;p,q,r]

disp ('Para sumar se realiza como si fueran dos variables:');
disp ('Suma = MatrizA + MatrizB');

if size (Matrix_a) == size (Matrix_b)
La_Suma_es = Matrix_a + Matrix_b
else
    disp ('El tamaño no es adecuado.')
end

%ok
end

% -----
function varargout = pushbutton5_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton5.
%disp('pushbutton5 Callback not implemented yet.')

%realizo restas
clc;
global a b c d e f g h0 i j k l m n o p q r;

if isempty(a) & isempty(b) & isempty(c) & isempty(d) & isempty(e) & isempty(f)
& isempty(g) & isempty(h0) & isempty(i) & isempty(j) & isempty(k) &
isempty(l) & isempty(m) & isempty(n) & isempty(o) & isempty(p) &
isempty(q) & isempty(r)

    run error6

```

---

---

```

else

disp ('Para definir una matriz solo hay que poner en la ');
disp ('linea de comandos: MatrixA = [(r1,c1),(r1,c2)];');
disp ('          [(r2,c1),(r2,c2)];');
disp ('o se puede usar el mismo renglon. ')
disp (' MatrixA = [(r1,c1),(r1,c2);(r2,c1),(r2,c2)]')

Matrix_a = [a,b,c;d,e,f;g,h0,i]
Matrix_b = [j,k,l;m,n,o;p,q,r]

disp ('Para restar se realiza como si fueran dos variables:');
disp ('Resta = MatrizA - MatrizB');

if size (Matrix_a) == size (Matrix_b)
    La_Resta_es = Matrix_a - Matrix_b

else
    disp ('El tamaño no es adecuado.')
end
%ok
end

% -----
function varargout = pushbutton6_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton6.
%disp('pushbutton6 Callback not implemented yet.')

%realizo Multiplicacion
clc;
global a b c d e f g h0 i j k l m n o p q r;

if isempty(a) & isempty(b) & isempty(c) & isempty(d) & isempty(e) & isempty(f)
& isempty(g) & isempty(h0) & isempty(i) & isempty(j) & isempty(k) &
isempty(l) & isempty(m) & isempty(n) & isempty(o) & isempty(p) &
isempty(q) & isempty(r)

    run error6

else
disp ('Para definir una matriz solo hay que poner en la ');
disp ('linea de comandos: MatrixA = [(r1,c1),(r1,c2)];');
disp ('          [(r2,c1),(r2,c2)];');
disp ('o se puede usar el mismo renglon. ')
disp (' MatrixA = [(r1,c1),(r1,c2);(r2,c1),(r2,c2)]')

Matrix_a = [a,b,c;d,e,f;g,h0,i]
Matrix_b = [j,k,l;m,n,o;p,q,r]

disp ('Para multiplicar se realiza como si fueran dos variables:');
disp ('Multiplica = MatrizA * MatrizB');

La_Multiplicacion_es = Matrix_a * Matrix_b

end

% -----
function varargout = pushbutton7_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton7.
%disp('pushbutton7 Callback not implemented yet.')

```

---

---

```

%realizo division
clc;
global a b c d e f g h0 i j k l m n o p q r;

if isempty(a) & isempty(b) & isempty(c) & isempty(d) & isempty(e) & isempty(f)
& isempty(g) & isempty(h0) & isempty(i) & isempty(j) & isempty(k) &
isempty(l) & isempty(m) & isempty(n) & isempty(o) & isempty(p) &
isempty(q) & isempty(r)

    run error6

else

disp ('Para definir una matriz solo hay que poner en la ');
disp ('linea de comandos: MatrixA = [(r1,c1),(r1,c2)];');
disp ('          [(r2,c1),(r2,c2)];');
disp ('o se puede usar el mismo renglon. ')
disp (' MatrixA = [(r1,c1),(r1,c2);(r2,c1),(r2,c2)]')

Matrix_a = [a,b,c;d,e,f;g,h0,i]
Matrix_b = [j,k,l;m,n,o;p,q,r]

disp ('Para dividir se realiza como si fueran dos variables:');
disp ('Division = MatrizA / MatrizB');

if size (Matrix_a) == size (Matrix_b)

La_Division_es = Matrix_a / Matrix_b

else
    disp ('El tamano no es adecuado.')
end

%ok
end

% -----
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton1.
%disp('pushbutton1 Callback not implemented yet.')

    close (gcbf)

% -----
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
%disp('pushbutton2 Callback not implemented yet.')

%texto
ponte = findobj ( gcbf, 'Tag' , 'text3' );

pon = 'Las operaciones con matrices, solo depende de las reglas básicas del
tamaño de las matrices a sumar, restar, dividir o multiplicar. Luego se realiza
la operación como si se tratase de dos variables, pero que ahora son matrices';

set ( ponte, 'String', pon );

%cambio estado del boton

```

---

---

```
cambio = findobj ( gcbf, 'Tag', 'inicio' );
set ( cambio , 'String' , 'Da enter' );
```

```
%ok
```

```
pause
```

```
pon2 = 'El procedimiento se observa en la ventana de comandos.';
```

```
set ( ponte , 'String' , pon2);
```

```
%cambio estado del boton
```

```
set ( cambio , 'String' , 'Inicio' );
```

```
%ok
```

```
% -----
```

### **Código Módulo Cálculo**

```
function varargout = deriva6(varargin)
```

```
% DERIVA6 Application M-file for deriva6.fig
```

```
% FIG = DERIVA6 launch deriva6 GUI.
```

```
% DERIVA6('callback_name', ...) invoke the named callback.
```

```
% Last Modified by GUIDE v2.0 22-Mar-2003 13:04:18
```

```
if nargin == 0 % LAUNCH GUI
```

```
    fig = openfig(mfilename,'reuse');
```

```
    % Generate a structure of handles to pass to callbacks, and store it.
```

```
    handles = guihandles(fig);
```

```
    guidata(fig, handles);
```

```
    if nargout > 0
```

```
        varargout{1} = fig;
```

```
    end
```

```
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
```

```
    try
```

```
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
```

```
switchyard
```

```
    catch
```

```
        disp(lasterr);
```

```
    end
```

```
end
```

```
% -----
```

```
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
```

```
% Stub for Callback of the uicontrol handles.pushbutton1.
```

```
%disp('pushbutton1 Callback not implemented yet.')
```

```
    close (gcbf)
```

```
% -----
```

```
function varargout = edit1_Callback(h, eventdata, handles, varargin)
```

```
% Stub for Callback of the uicontrol handles.edit1.
```

```
%disp('edit1 Callback not implemented yet.')
```

---

---

```
%defino funcion

global x funcion;

syms x;

funcion = eval (get (gcbo, 'String' ) );

% -----
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton3.
%disp('pushbutton3 Callback not implemented yet.')

%realizo calculos
clc;
global x funcion;

syms x;

if isempty(funcion)
    run error6
else
disp ('Para derivar priemero creamos la ecuacion:');

ecuacion = funcion

disp ('Ahora hacemos la primera derivada:');
disp ('diff (ecuacion)');

Primera_Derivada = diff (funcion)

disp ('o sea:');

pretty (Primera_Derivada)

disp ('Calculamos la segunda derivada:');
disp ('diff (ecuacion,2)');

Segunda_Derivada = diff (funcion,2)

disp ('o sea:');

pretty (Segunda_Derivada)

disp ('Finalmente se calcula la tercera derivada:');
disp ('diff (ecuacion,3)');

Tercera_Derivada = diff (funcion,3);

disp ('o sea:');

pretty (Tercera_Derivada)

end
```

---

---

```

% -----
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
%disp('pushbutton2 Callback not implemented yet.')

%en el boton de inicia

ponte = findobj ( gcbf, 'Tag' , 'text3' );

pon = 'En MatLab obtener la derivada de cualquier función se realiza con el
comando diff. ';

set ( ponte, 'String', pon );

%cambio estado del boton

cambio = findobj ( gcbf, 'Tag', 'inicio' );
set ( cambio , 'String' , 'Da enter' );

%ok

pause

pon2 = ' Observe en la ventana de comandos.';

set ( ponte , 'String' , pon2);

%cambio estado del boton

set ( cambio , 'String' , 'Inicio' );

%ok

% -----

```

### **Código Módulo Ecuaciones Diferenciales**

```

function varargout = edopc6(varargin)
% EDOPC6 Application M-file for edopc6.fig
% FIG = EDOPC6 launch edopc6 GUI.
% EDOPC6('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 29-Aug-2002 23:20:18

if nargin == 0 % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);

    if nargin > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK

    try
        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
    catch
        switchyard
    catch

```

---

---

```

                disp(lasterr);
            end
        end

end

% -----
function varargout = edit1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit1.
%obtengo ecuacion

global la_ecuacion1 ;
% no es necesario definir variables, por default estan Df f t
la_ecuacion1 = eval ( get ( gcbo, 'String' ) );

% -----
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton1.
%salgo

close (gcbf)

% -----
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
%texto

ponte = findobj ( gcbf, 'Tag' , 'text3' );
pon = 'Otro uso de dsolve, es usando condiciones iniciales. ';
set ( ponte, 'String', pon );

%cambio estado del boton

cambio = findobj ( gcbf, 'Tag', 'inicio' );
set ( cambio , 'String' , 'Da enter' );

pause

pon2 = 'Observe en la ventana de comandos,dsolve (ecuacion, condicion).';
set ( ponte , 'String' , pon2);

%cambio estado del boton

set ( cambio , 'String' , 'Inicio' );

%ok

% -----
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton3.

%realizo calculos
clc;
global la_ecuacion1 la_condicion01;

```

---

---

```

if isempty(la_ecuacion1) | isempty(la_condicion01)
    run error6
else
    disp ('Para resolver una ecuacion siferencial con condiciones iniciales');
    disp ('Escribimos la ecuacion:');

    Mi_ecuacion = la_ecuacion1
    disp ('Escribimos la condicion');
    Mi_condicion = la_condicion01
    disp ('Aplicamos: dsolve ( Mi_ecuacion ,condicion1 )' );
    solucion = dsolve ( Mi_ecuacion, Mi_condicion )
    pretty (solucion)
end

```

```

% -----
function varargout = edit3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.edit3.
%disp('edit3 Callback not implemented yet.')

global la_condicion01 ;
% no es necesario definir variables, por default estan Df f t
la_condicion01 = eval ( get ( gcbo, 'String' ) );

% -----

```

### **Código Módulo Programación Básica**

```

function varargout = var6(varargin)
% VAR6 Application M-file for var6.fig
% FIG = VAR6 launch var6 GUI.
% VAR6('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 06-Jun-2003 21:08:23

if nargin == 0 % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);

    if nargin > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK

```

---



---

```

        try
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
switchyard
        catch
            disp(lasterr);
        end
end

% -----
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton2.
%disp('pushbutton2 Callback not implemented yet.')

close (gcbf)

% -----
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton3.
%disp('pushbutton3 Callback not implemented yet.')

%el inicia
clc;
%para el texto
boton_ok = findobj ( gcbf, 'Tag' , 'el_ok' );
boton_ok1 = findobj ( gcbf , 'Tag', 'eje2');
boton_ar1 = findobj (gcbf, 'Tag', 'ar1');
boton_ar2 = findobj (gcbf, 'Tag', 'ar2');

%deshabilito
set ( boton_ok, 'Enable', 'Off' );
set ( boton_ok1, 'Enable', 'Off');
set (boton_ar1, 'Enable', 'Off');
set (boton_ar2, 'Enable', 'Off');

ponte = findobj ( gcbf, 'Tag' , 'text1' );

pon = 'Hay dos tipos de variables: numericas y caracter. ';

set ( ponte, 'String', pon );

%cambio estado del boton

cambio = findobj ( gcbf, 'Tag', 'inicio' );
set ( cambio , 'String' , 'Da enter' );

%ok

pause

pon2 = 'Para pedir una cadena tipo caracter usamos syms variable. Que junto con
input (pregunta, varialbe) podemos pedir datos desde la ventana de comandos ';

set ( ponte, 'String', pon2 );

set ( cambio , 'String' , 'Da enter' );

pause

%ok2

```

---

---

```
pon3 = 'Para observar como pedir datos da clic en Ejemplo 1.Para datos numericos
da clic en Ejemplo 2 ';
```

```
set ( ponte, 'String', pon3 );
```

```
set ( cambio , 'String' , 'Inicia' );
```

```
%para boton
```

```
set ( boton_ok, 'Enable', 'On' );
```

```
set ( boton_ok1, 'Enable', 'On');
```

```
set (boton_ar1,'Enable', 'On');
```

```
set (boton_ar2, 'Enable', 'On');
```

```
% -----
function varargout = el_ok_Callback(h, eventdata, handles, varargin)
```

```
% Stub for Callback of the uicontrol handles.el_ok.
```

```
%disp('el_ok Callback not implemented yet.')
```

```
%para ejemplo 1
```

```
disp ('Primero creamos la variable:(Debe usarse s por string)');
```

```
disp ( '      syms s');
```

```
disp ( 'creamos el destino y pedimos el dato:');
```

```
disp ('Nota: la pregunta y la variable deben estar entre ');
```

```
disp('      comilla simple ');
```

```
disp ( '      mitexto = input (quien es>, s) ' );
```

```
disp ('ponemos el dato:');
```

```
disp ( '      disp (mitexto)' );
```

```
disp ( '- Da enter para probar -');
```

```
pause
```

```
run pideej1
```

```
% -----
function varargout = pushbutton5_Callback(h, eventdata, handles, varargin)
```

```
% Stub for Callback of the uicontrol handles.pushbutton5.
```

```
%disp('pushbutton5 Callback not implemented yet.')
```

```
%para ejemplo 2
```

```
disp ('En casos numericos, no es necesario definir el tipo');
```

```
disp ( 'Al igual que anteriormente creamos el destino ');
```

```
disp ( ' y pedimos el dato:');
```

```
disp ('Nota: la pregunta deben estar entre ');
```

```
disp('      comilla simple ');
```

```
disp ( '      midato = input (da un numero>) ' );
```

```
disp ('Otra forma de poner el dato:( %d entre comillas simples)');
```

```
disp ( '      sprintf ( %d, midato)' );
```

```
disp ( '- Da enter para probar -');
```

```
pause
```

```
run pideeje2
```

```
% -----
function varargout = pushbutton6_Callback(h, eventdata, handles, varargin)
```

---

---

```
% Stub for Callback of the uicontrol handles.pushbutton6.
%disp('pushbutton6 Callback not implemented yet.')
%archivo 1

edit pideej1

% -----
function varargout = pushbutton7_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pushbutton7.
%disp('pushbutton7 Callback not implemented yet.')
%archivo 2

edit pideeje2
```

**Nota:**

El código mostrado aquí solo corresponde a los módulos más importantes del software, debido a que son más de 50 hojas de código es imposible imprimir todos los archivos.

---