PRÁCTICA 08.- GUIDE

Calculadora

Montor Láscares Pedro Antonio
Ortiz Rosas Mario

Contenido

Capítulo 1 2
Introducción 2
Capítulo 2 3
Marco Teórico 3
2.2 Matlab 3
2.3 Guide 3
Capítulo 3 5
Desarrollo y resultados 5
3.1 Programación en GUIDE 5
Capítulo 4 15
Conclusiones 15
Bibliografía 16

Capítulo 1

Introducción

Las interfaces gráficas de usuario (GUI-Graphical User Interface en inglés), es la forma en que el usuario interactúa con el programa o el sistema operativo de una computadora. Una GUI contien diferentes elementos gráficos tales como: botones, campos de texto, menús, gráficos, etc.

Existen diferentes lenguajes de programación que permiten crear una GUI tales como: C, Visual Basic, TK, etc. Solo por mencionar algunos. Todos ellos perimiten usar una manera muy sencilla usando GUIDE (Graphical User Interface Developmet Enviroment).

Capítulo 2

Marco Teórico

Aquí se presentará la teoría de una manera reducida de los elementos que conforman nuestro sistema, para información más detallada se puede consultar la bibliografía.

2.2 Matlab

MATLAB es un acrónimo de *Matrix Laboratory*. Actualmente es uno de los entornos de programación más usados en el campo académico como en la industria para la realización de cálculos poderosos y creación de interfaces.

El lenguaje de programación proporciona un sencillo acceso a algoritmos numéricos que incluyen matrices, considerado como un lenguaje de programación matemático de alto nivel integrado en un entorno gráfico amigable, visualización de datos, funciones, gráficas 2D y 3D, procesamientos de imágenes, video, computación numérica, etc. Particularmente, en ingeniería es una herramienta muy poderosa para realizar aplicaciones en mecatrónica, robótica, control y automatización. [2]

Figura 2.1: Logo de Matlab

2.3 Guide

La interfaz gráfica de usuario de MATLAB recibe el nombre de Graphical User Interface Development Enviroment (GUIDE), es la manera en que el usuario puede interactuar con un programa de forma sencilla, de tal forma que permite trabajar intuitivamente. Contiene diferentes elementos gráficos tales como: botones, barras deslizantes, campo de texto, menús, gráficos, inspector de propiedades, etc.

Guide proporciona un entorno gráfico al usuario que facilita el diseño de aplicaciones por medio de bloques para representar enlaces dinámicos generando una interfaz gráfica al usuario para interactuar con el programa.

Cuando se inicia el ambiente GUIDE se obtiene una ventana como la que se muestra a continuación, hay dos opciones de trabajo sobre esta ventana inicial: una puede ser para crear una nueva aplicación GUI o abrir alguna ya existente.

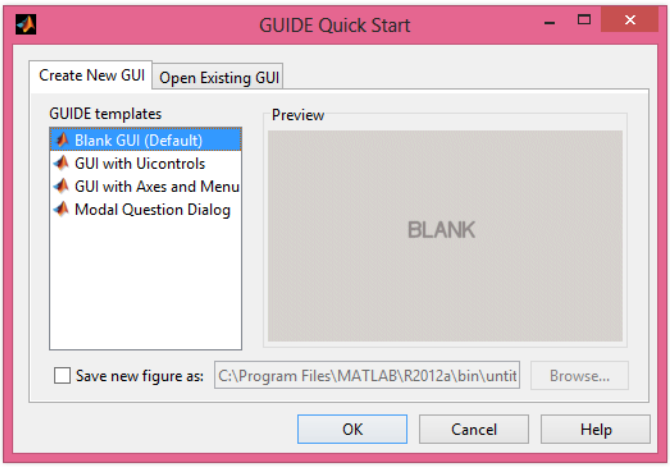


Figura 2.2: Inicio del entorno GUIDE.

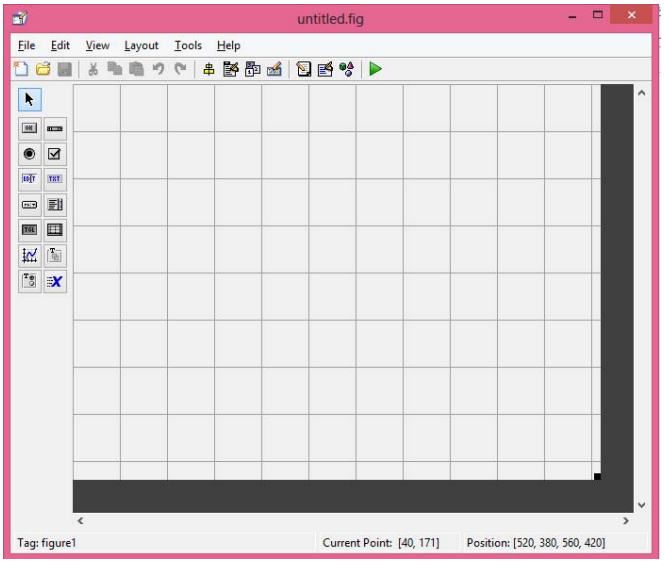


Figura 2.2: Sesión nueva en blanco.

Capítulo 3

Desarrollo y resultados

3.1 Programación en GUIDE

Para crear una calculadora, en este caso sería un GUI en Matlab usamos GUIDE ya sea que tecleemos guide en la ventana de comandos de Matlab o lo ejecutemos del menú principal File -> New -> GUI.

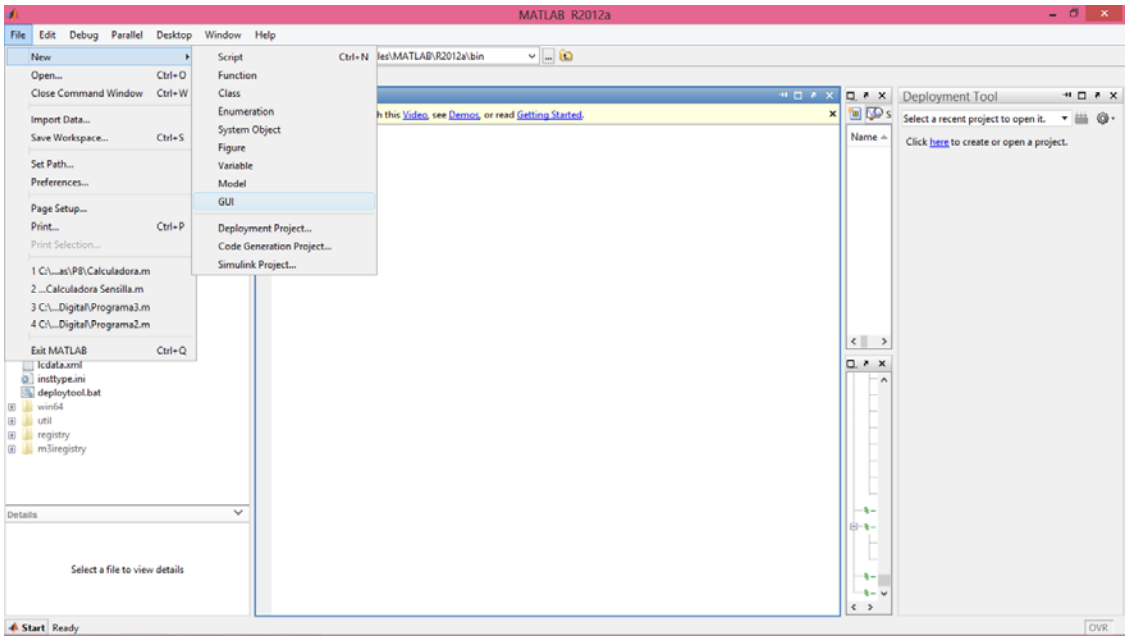
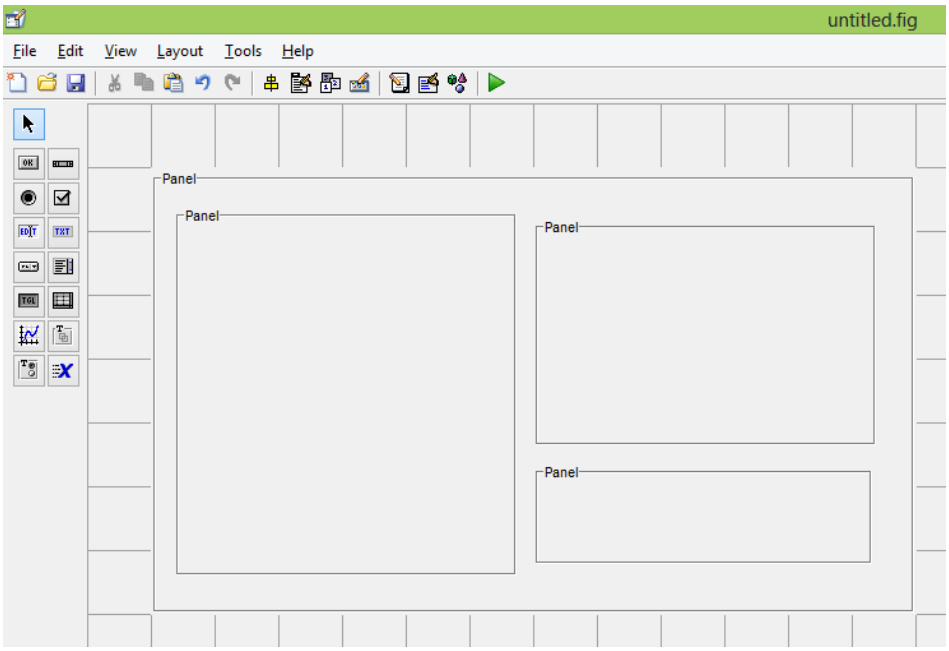


Figura 3.1: Ejecución de GUIDE usando el menú principal o la ventana de comandos.

Una vez hecho lo anterior MatLab nos mostrará un área de diseño similar a la figura 2.2. En la parte superior se encuentran los menús y opciones de GUIDE, en la parte izquierda se aprecian los diferentes controles y en la parte central el área de diseño donde pondremos los controles a usar.

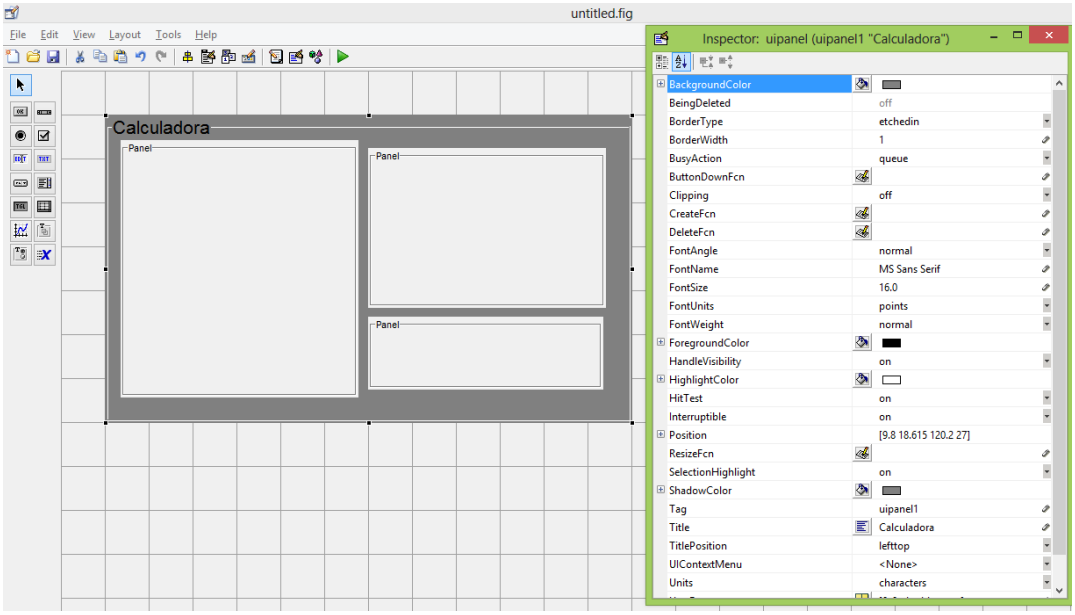
Una vez abierto un nuevo proyecto, se procede con el diseño de la interfaz.

Paso 1 *Áreas de trabajo*.- De la ventana de componentes del lado izquierdo, Seleccionar panel y se dibujan cuatro cuadros, estos para delimitar las áreas.



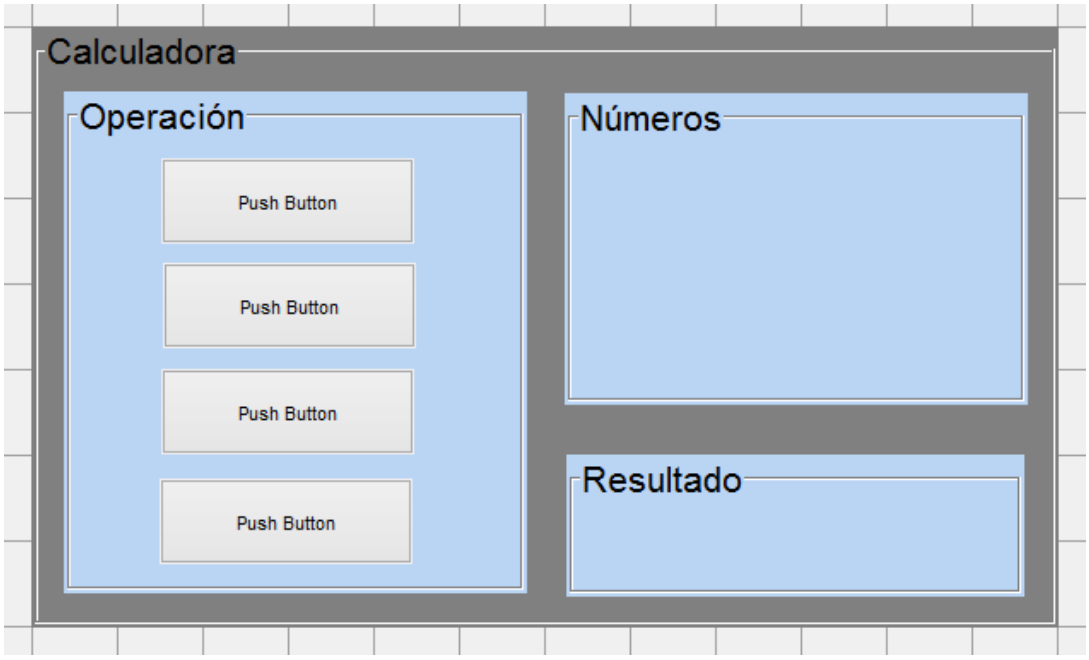
Para cambiar las propiedades de cada panel se selecciona -> botón derecho -> Property Inspector, y en la ventana que aparece se modifican estos tres parámetros:

- BackgroundColor : Cambio de color (Gris oscuro).
- Title: Nombre del panel (Calculadora).
- FontSize: Tamaño de lentra (16).



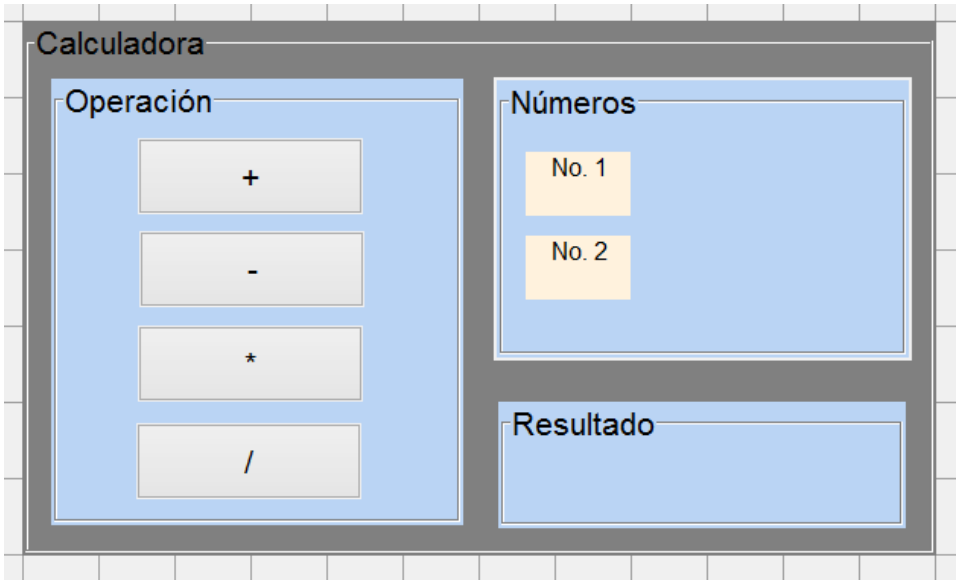
De igual manera se modifican los demás paneles para quedar de la siguiente forma:

Paso 2 *Botones de operación*.- De la ventana de componentes, seleccionar Push Button e insertar cuatro en el área de operación.



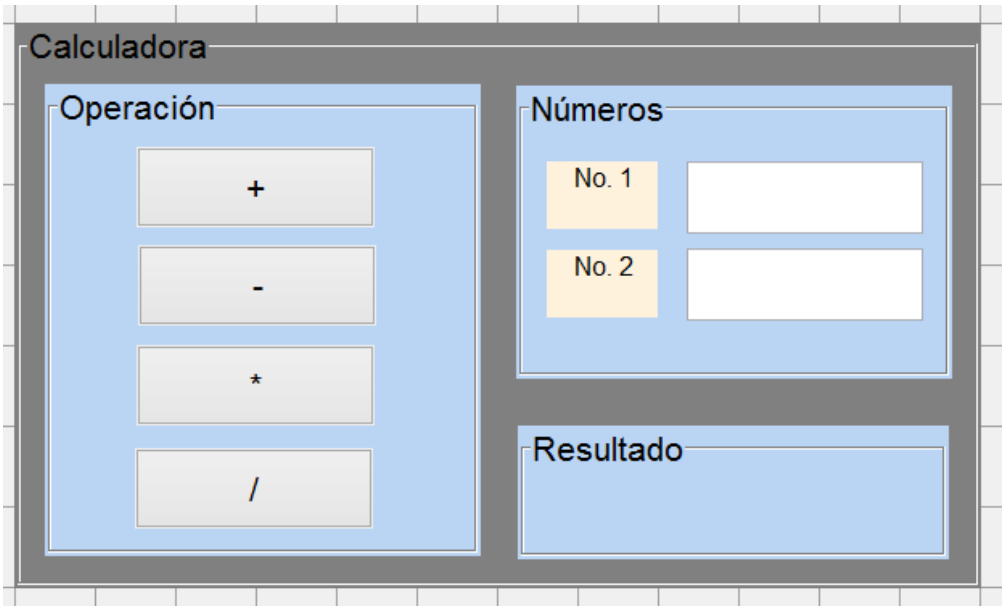
Para editar las propiedades de los botones en exactamente como en la edición de los paneles, en este caso solo se modifica en nombre del botón, se cambia String por el símbolo de cada operación y el tamaño del símbolo (FontSize).

Paso 3 *Texto de identificación*.- Seleccionar el Static Text e insertar dos en el área de Números. Estos son para identificar los números a operar, así que solo se edita en las propiedades el String por la etiqueta No.1 y No.2, quedando de la siguiente forma.

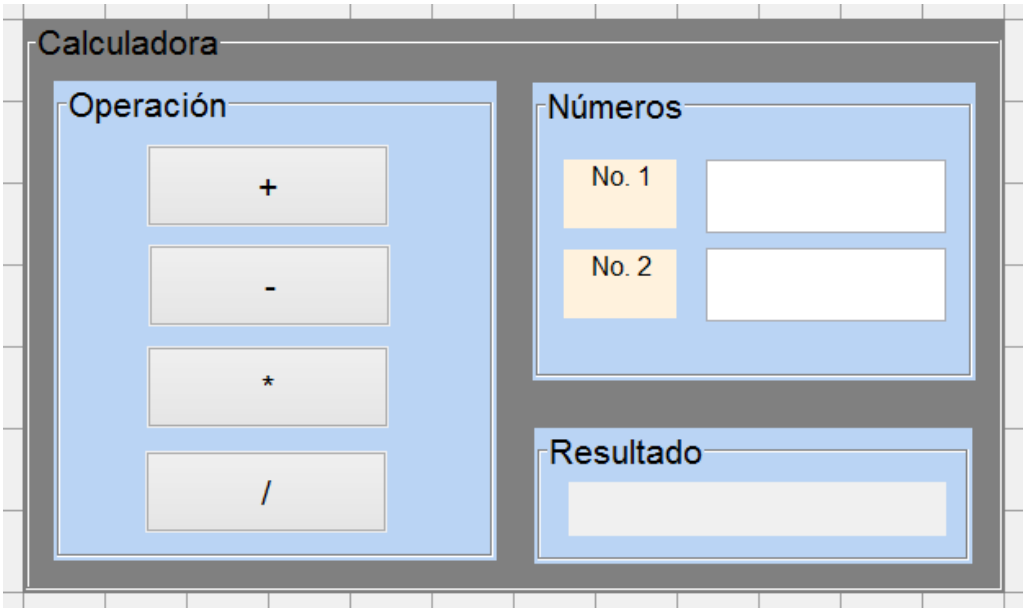


Paso 4 *Campos de números*.- Seleccionar el Edit Text e insertar dos en el área de Números. Estos campos serán donde el usuario podrá escribir los números a operar.

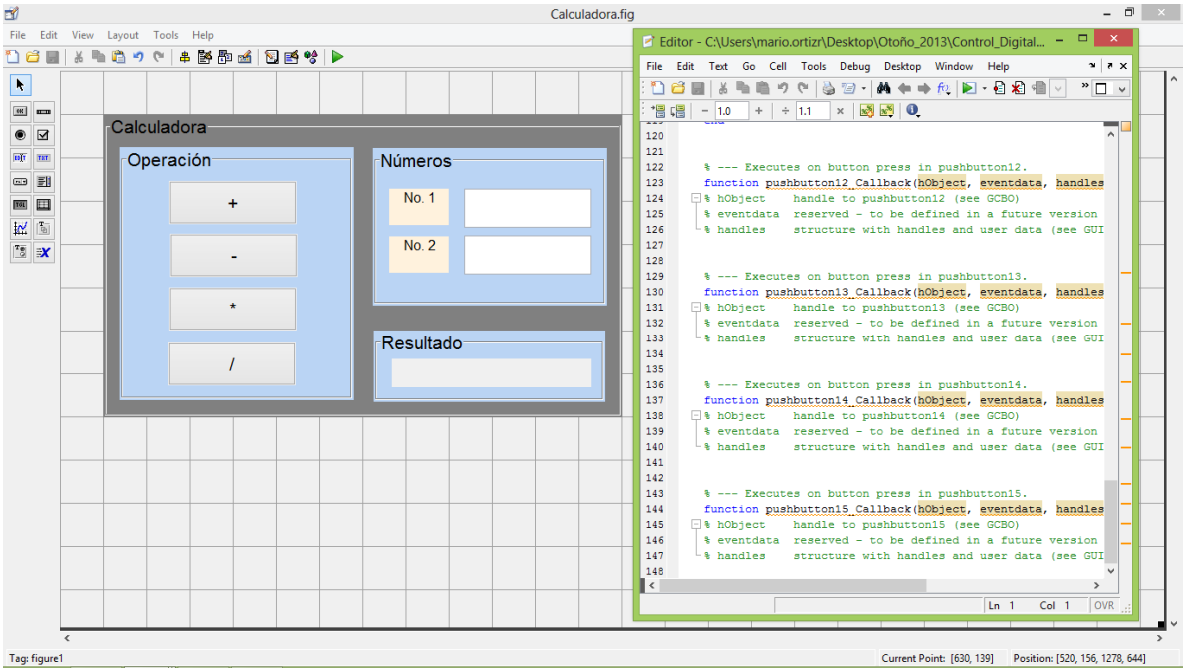
En propiedades se tiene que borrar el texto por default en el String.



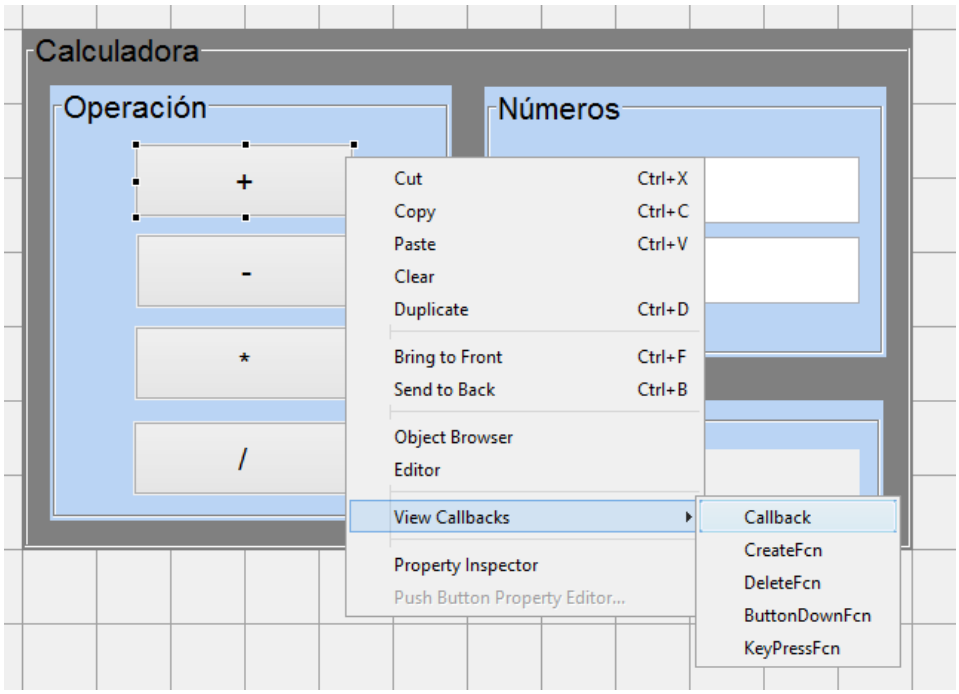
Paso 5 *Muestra de resultado*.- Se selecciona Static Text y se inserta en el área de resultado, modificando las propiedades para borrar el texto de default (String).

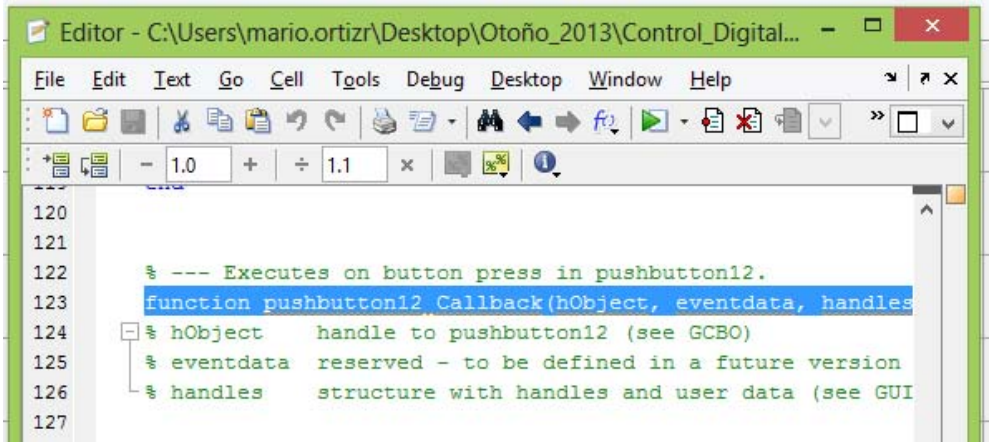


Paso 6 *Programación de botones*.- Se guarda el proyecto con el nombre de Calculadora, automáticamente se abre el archivo con extensión .m donde se programará la función de los botones.



Una vez abierto el código, seleccionamos el Push Button suma -> click derecho -> View Callbacks -> Callback. Estos pasos indicará en que parte del código se podrá hacer la programación de este Push Button.





Una vez ubicados, se programa la función suma.

```
% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
a = str2num(get(handles.edit2, 'String'));
b = str2num(get(handles.edit3, 'String'));
c = a + b;
set(handles.text4,'String',c);
```

Nota: La definición de edit2 y edit3 se define corroborando el nombre, es decir, en el campo de números (Edit text), con la ventana de propiedades en la descripción Tag dice el nombre, éste es el que se esribe. En caso de A seria el Campo designado para el primer número mientras que el en B sería en campo designado para el segundo número a operar.

En el caso de text 4, este es el nombre del Static Text donde será mostrado el resultado, para ello se corrobora el nombre en propiedades en la descripción Tag.

El nombre de los botones dependerá del orden en que fue dibujada la interfaz, por ello en necesario corroborar el nombre de cada componente.

Ya que se programó la función suma, será de igual forma para las demás, solo cambiando la operación (Incluso se puede hacer solo un copy-paste y solo cambiar la operación).

```
132
133     % --- Executes on button press in pushbutton13.
134     function pushbutton13_Callback(hObject, eventdata, handles)
135     % hObject    handle to pushbutton13 (see GCBO)
136     % eventdata  reserved - to be defined in a future version of MATLAB
137     % handles    structure with handles and user data (see GUIDATA)
138 -   a = str2num(get(handles.edit2, 'String'));
139 -   b = str2num(get(handles.edit3, 'String'));
140 -   c = a - b;
141 -   set(handles.text4,'String',c);
142
143
144     % --- Executes on button press in pushbutton14.
145     function pushbutton14_Callback(hObject, eventdata, handles)
146     % hObject    handle to pushbutton14 (see GCBO)
147     % eventdata  reserved - to be defined in a future version of MATLAB
148     % handles    structure with handles and user data (see GUIDATA)
149 -   a = str2num(get(handles.edit2, 'String'));
150 -   b = str2num(get(handles.edit3, 'String'));
151 -   c = a * b;
152 -   set(handles.text4,'String',c);
153
154
155     % --- Executes on button press in pushbutton15.
156     function pushbutton15_Callback(hObject, eventdata, handles)
157     % hObject    handle to pushbutton15 (see GCBO)
158     % eventdata  reserved - to be defined in a future version of MATLAB
159     % handles    structure with handles and user data (see GUIDATA)
160 -   a = str2num(get(handles.edit2, 'String'));
161 -   b = str2num(get(handles.edit3, 'String'));
162 -   c = a / b;
163 -   set(handles.text4,'String',c);
164
```

Solo basta guardar el proyecto y la calculadora se puede operar.

Paso 7 *Prueba*.- Se ejecuta la interfaz y se prueban valores.

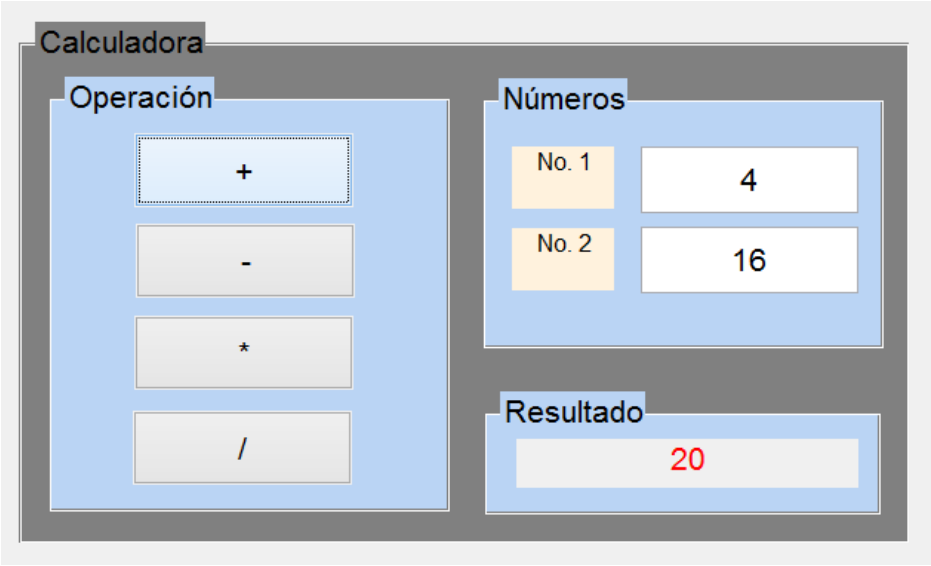


Figura 3.1.7.1: Prueba suma.

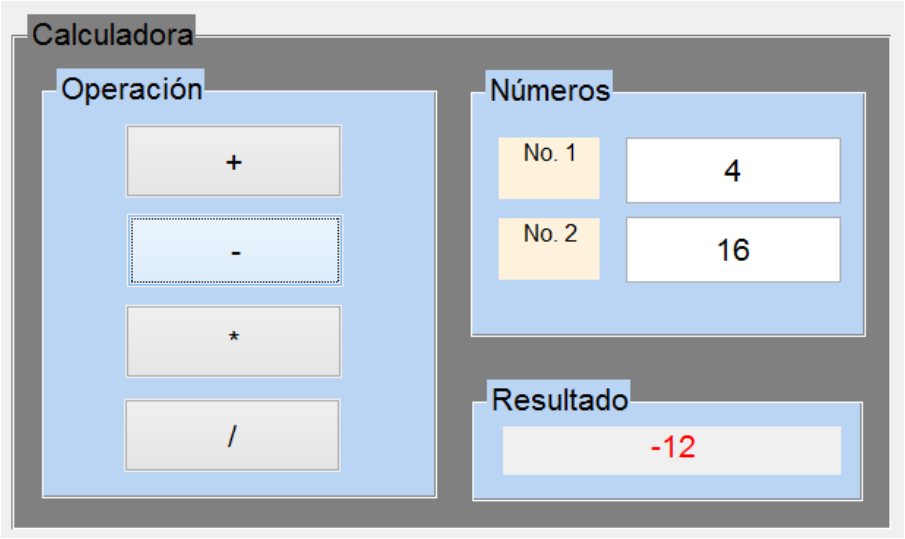


Figura 3.1.7.2: Prueba resta.

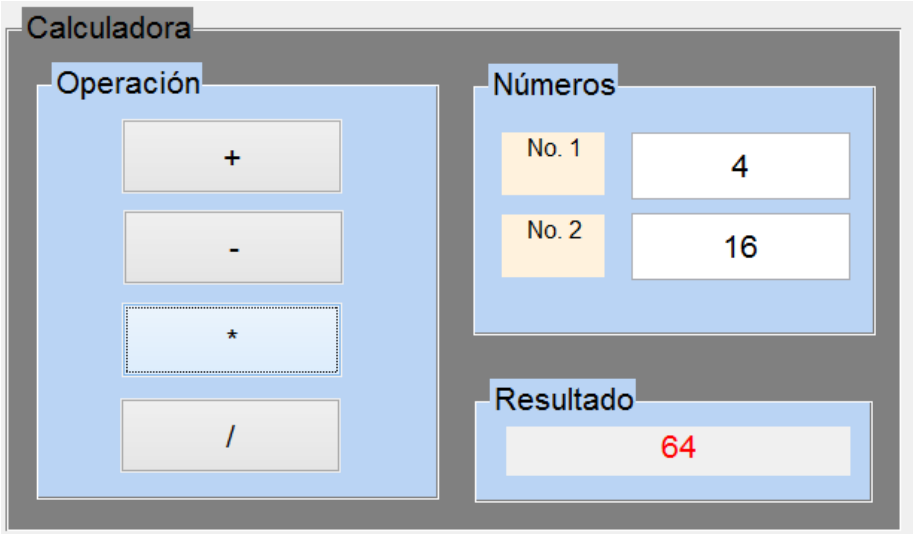


Figura 3.1.7.3: Prueba multiplicación.

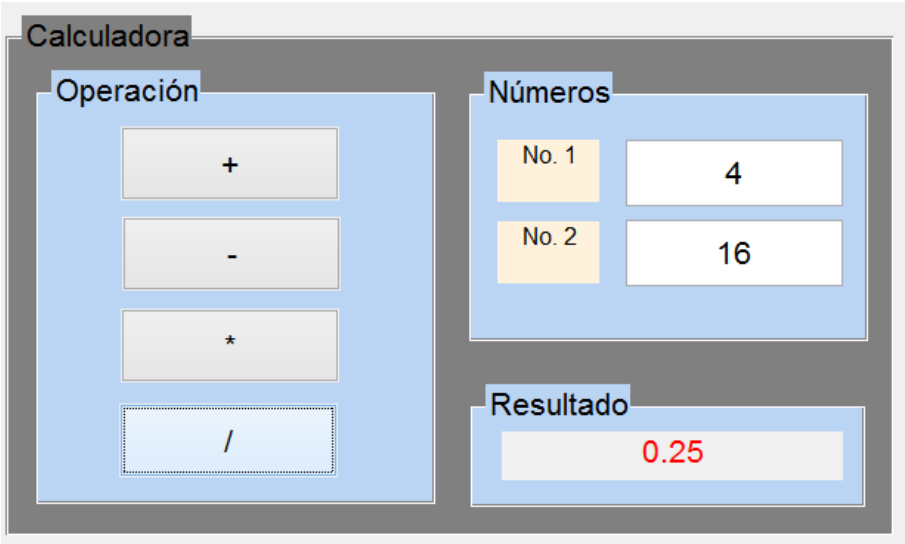


Figura 3.1.7.4: Prueba división.

Capítulo 4

Conclusiones

Matlab es un software que permite el desarrollo de programación junto con una interfaz gráfica la cual facilita a un usuario el uso de un programa. El ambiente de programación GUIDE facilita la edición de texto y colores para personalizar una interfaz y que esta sea amigable con el usuario. Además de ser fácil de usar y programar. Esta práctica de calculadora es un ejemplo básico para comprender y familiarizarse con esta función que te proporciona Matlab, esto permitirá más adelante desarrollar interfaces más completas y con mayes funciones, incluso la posibilidad de ejecutarla con hardware.

Bibliografía

[1] Ogata, Katsuhiko (1996). *Sistemas de Control en Tiempo discreto*. 2da. Edición. Editorial Prentice Hall Hispanoamericana S. A.
[2] Reyes Cortés, F.; Cid Monjaraz, J. & Vargas Soto, E. (2013). *Mecatrónica, Control y Automatización*. Alfaomega: México.
[3] Reyes Cortés, F. (2012). *MATLAB Aplicado a Robótica y Mecatrónica*. Alfaomega: México.