

7.5 Ejemplos

Empezamos esta sección con una GUI sencilla y luego procedemos con ejemplos que involucran más componentes de las GUIs.

EJEMPLO 7.1 Una GUI sencilla

Para ilustrar el proceso de creación de una GUI consideremos una muy simple. Quizá uno de los ejemplos más simples es el de Hola Mundo en cualquier lenguaje de programación. Para ello creamos una nueva figura y agregamos un Static Text y un Push Button (ver figura 7.5) y lo guardamos como HolaMundo.fig; la extensión fig la da MATLAB automáticamente.

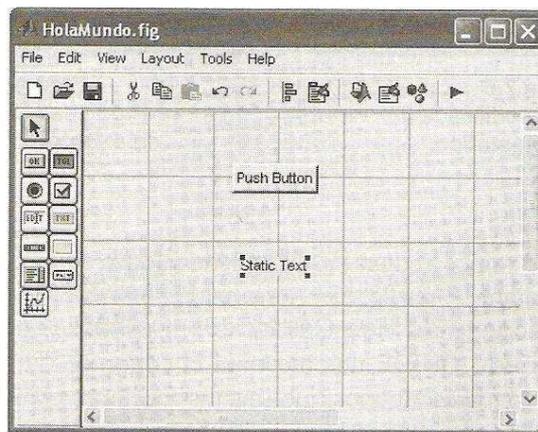


Figura 7.5 Ventana de diseño de la GUI con el **Texto estático** y el **Botón de presión**.

Después de guardar nuestro archivo, GUIDE automáticamente nos mostrará una ventana nueva con el código del archivo `HolaMundo.m` generado por GUIDE. Si se observa el código se puede notar que hay líneas con %, lo cual indica que se trata de un comentario en el archivo script. Ahora en la ventana de GUIDE seleccionamos el Static Text y activamos el Inspector de Propiedades para modificar la propiedad String por Hola Mundo (figura 7.6). Se repite la misma operación para el Botón de presión pero en String escribimos Salir (figura 7.7)

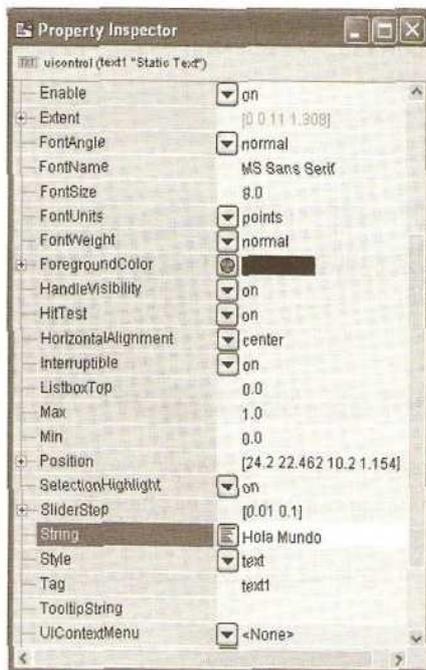


Figura 7.6
Propiedad modificada
en el Texto estático.



Figura 7.7
Propiedad modificada
en el Botón de presión.

Como siguiente paso vamos a editar el callback de nuestro botón Salir para que al ser activado cierre la ventana. Para ello seleccionamos el botón Salir con el botón derecho del apuntador y seleccionamos View Callbacks-> Callback como se muestra en la figura 7.8. Al hacer esto GUIDE nos mostrará el editor de funciones del archivo `HolaMundo.m` en donde escribimos nuestras instrucciones como se muestra en la figura 7.9. Para cerrar una figura en MATLAB la forma más sencilla es agregar

```
close (gcbf)
```

lo cual indica que cerraremos una figura de tipo gráfico. La manera de agregarlo en nuestra función se aprecia en la figura 7.9. Una vez guardado el archivo HolaMundo.m activamos la figura con la herramienta Activación de GUI y tendremos una salida como se muestra en la figura 7.10.

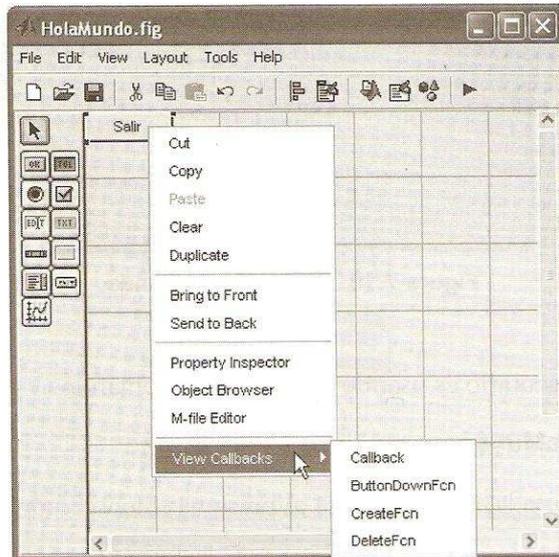


Figura 7.8 Editando el callback del botón Salir.

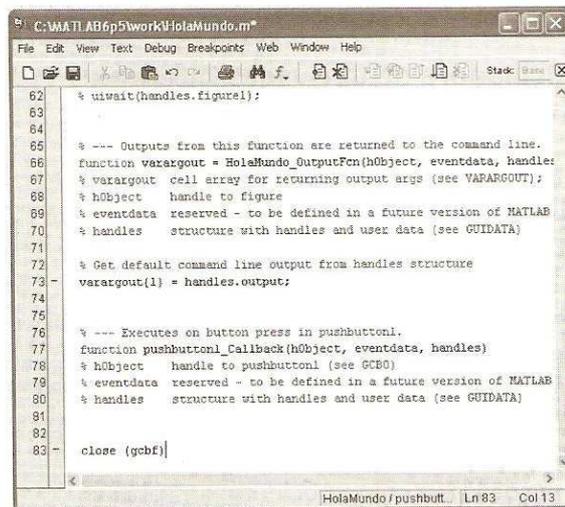


Figura 7.9 Agregando código a la función del botón Salir.

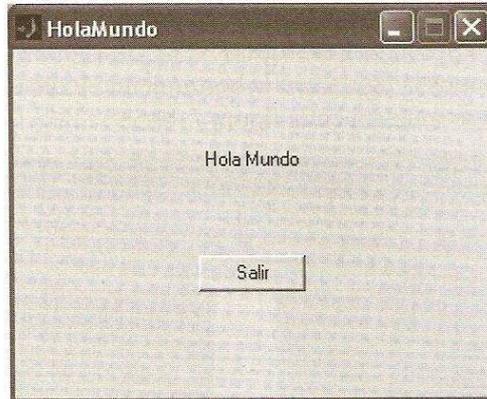


Figura 7.10 GUI de Hola Mundo.

Otra manera de ejecutarlo es usando la ventana de comandos y escribiendo:

```
EDU>>HolaMundo
```

Este GUI sólo ejecuta la acción de Salir al presionar dicho botón.

Hay que mencionar que en este momento contamos con dos archivos: HolaMundo.fig y HolaMundo.m. Ambos archivos son necesarios para poder ejecutar nuestro GUI.

EJEMPLO 7.2 Calculadora de derivadas/integrales/Fourier (Calculador DEINFOU)

Nuestro segundo ejemplo consiste en la creación de una GUI para el cálculo de derivadas, integrales y evaluación de la transformada de Fourier. Este proyecto nos permitirá explicar cómo usar los Cuadros de Edición (Edit Text) para pedir información y los Static Text para desplegar resultados, así como algunas otras propiedades importantes. Para empezar creamos una nueva figura y agregamos un Static Box, a un lado ponemos un Edit Text y al lado del Edit Text ponemos un Push Button, debajo de estos controles ponemos seis Static Text y finalmente agregamos dos Push Buttons y lo guardamos como CalcDelntFour.fig. Al finalizar tendremos una interfase como la que se muestra en la figura 7.11. El fondo blanco de nuestro GUI se puede lograr colocando el cursor sobre cualquier punto del GUI, haciendo doble pulsación con el botón izquierdo para que se abra el Editor de Propiedades y ahí seleccionamos que el color sea blanco.

Ahora en el primer Static Text cambiamos sus propiedades con el Inspector de Propiedades y modificamos el String a Funcion, como se muestra en la figura 7.12. En Text Box cambiamos la propiedad de String a x, al primer Push Button le cambiamos el String a Calcula.

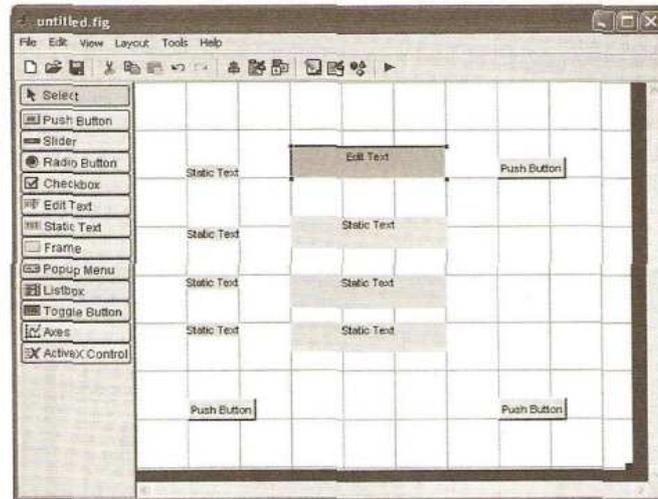


Figura 7.11 Interfase del Proyecto CalcDEINFOU.

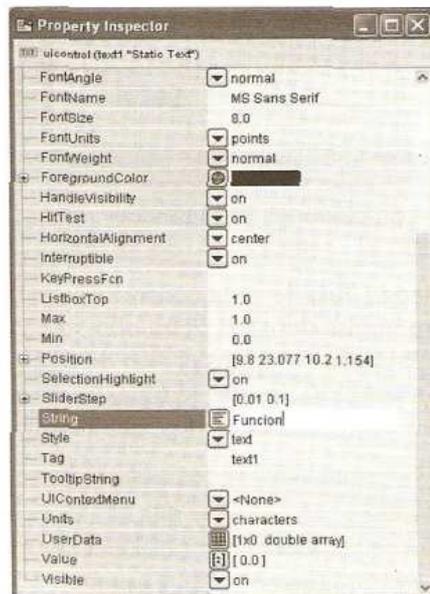


Figura 7.12
Cambio de la propiedad
String a Funcion.

En el siguiente orden modificamos las propiedades de String de cada Static Text restante como:

String: Derivada
 String: Integral
 String: Fourier

Para los Static Text abajo del Edit Text no escribimos nada para que queden en blanco.

Finalmente para el Push Button del lado izquierdo modificamos la propiedad String a Limpia y para el Push Button del lado derecho a Salir. Finalmente obtendremos los controles como se muestra en la figura 7.13.

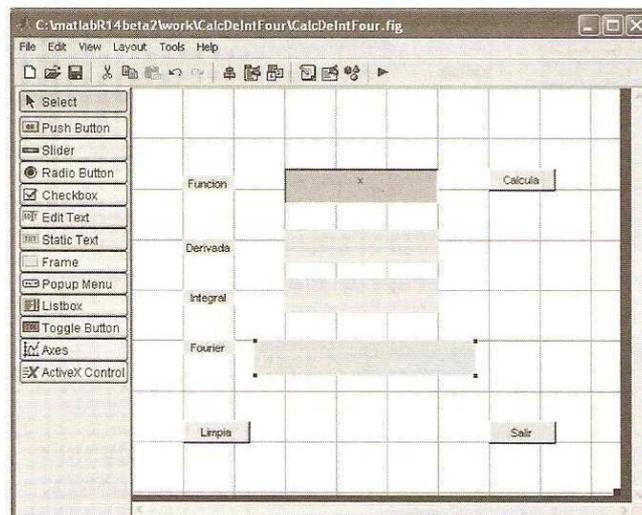


Figura 7.13 GUI con propiedades String modificadas.

El siguiente paso es editar la propiedad Tag de cada control. El Tag es el nombre como se va a identificar cada componente del GUI. Para ello seleccionamos Tag en el Inspector de propiedades. Primero editamos el Tag de nuestro control de la caja de texto (Edit Text) por la *ecuación*, lo que se muestra en la figura 7.14. Para los Static Text que no tienen texto modificamos sus Tag en el siguiente orden por:

resultado_derivada
 resultado_integral
 resultado_Fourier

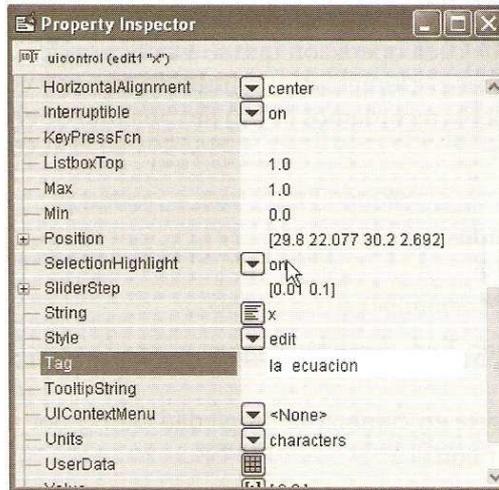


Figura 7.14
Propiedad Tag modificadas
en el Edit Text.

El Tag de los controles restantes es igual a su String correspondiente.

Ahora vamos a editar el callback del botón Salir y agregamos el código necesario para cerrar la ventana. El código para salir es

```
close(gcf);
```

y después de agregar `close(gcf)`, el código del botón salir queda como se muestra en la figura 7.15.

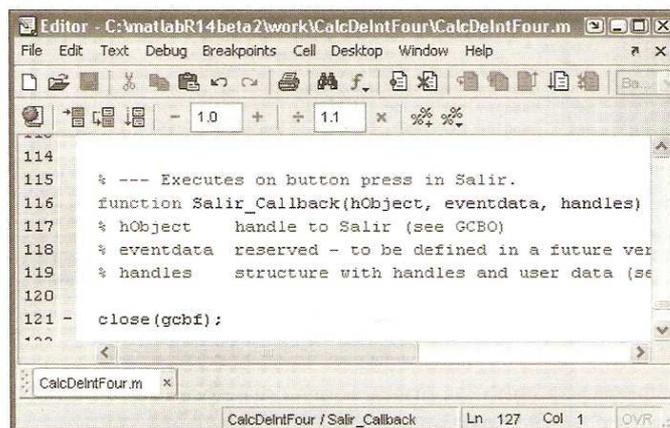


Figura 7.15 Código completo del botón Salir.

Luego editamos el callback del botón Limpia, cuya acción es limpiar el Edit Text de la función y los resultados de cada operación que se escriben en los Static Text. Para limpiar necesitamos conocer el identificador o “handle” de cada componente a limpiar. Esto lo podemos saber si recordamos el Tag de cada uno de ellos. De esta manera los identificadores son

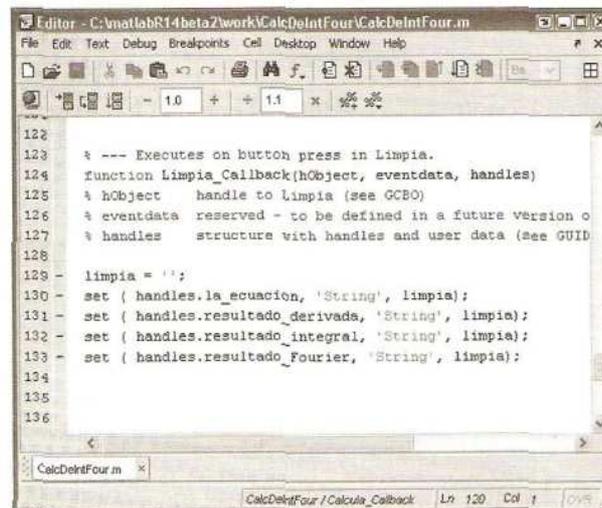
```
handles.la_ecuacion
handles.resultado_derivada
handles.resultado_integral
handles.resultado_Fourier
```

para el Edit Text y los tres Static Text, respectivamente.

El comando para limpiar consiste en cambiar la propiedad String por una cadena en blanco. Esto se puede hacer con

```
limpia = '';
set ( handles.resultado_derivada, 'String', limpia),
```

Proseguimos a limpiar los otros dos Static Text y el Edit Text de la misma manera. El código del botón limpiar queda como se muestra en la figura 7.16.



```

122
123 % --- Executes on button press in Limpia.
124 function Limpia_Callback(hObject, eventdata, handles)
125 % hObject    handle to Limpia (see GCBO)
126 % eventdata  reserved - to be defined in a future version o
127 % handles    structure with handles and user data (see GUID
128
129 - limpia = '';
130 - set ( handles.la_ecuacion, 'String', limpia);
131 - set ( handles.resultado_derivada, 'String', limpia);
132 - set ( handles.resultado_integral, 'String', limpia);
133 - set ( handles.resultado_Fourier, 'String', limpia);
134
135
136

```

Figura 7.16 Código del botón Limpia.

Finalmente llevaremos a cabo las operaciones correspondientes que se ejecutan al presionar el botón Calcula. Para ello editamos el callback de este botón según el siguiente algoritmo:

1. Obtengo la ecuación usando el identificador
2. Hago cálculos
3. Presento resultados

Para obtener la ecuación que se escriba en el Edit Text usamos

```
ecuacion = eval ( get ( handles.la_ecuacion, 'String'));
```

Para realizar el cálculo de la derivada:

```
derivada1 = diff (ecuacion);
```

Ahora preparo el resultado para desplegarlo en forma de caracter usando la instrucción char():

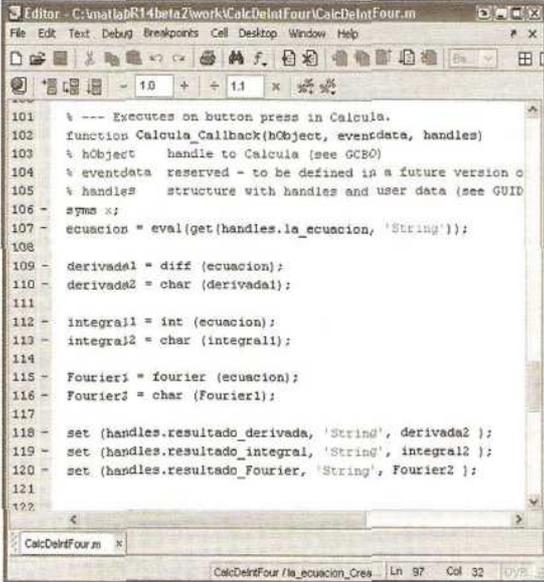
```
derivada2 = char (derivada1);
```

Finalmente despliego el resultado de la derivada:

```
set (handles.resultado_derivada, 'String', derivada2 );
```

Y realizo la misma operación para los otros dos cálculos. En la figura 7.17 se aprecia el código completo del botón Calcula.

Para probar nuestra aplicación activamos la figura y tendremos una salida como en la figura 7.18 para la función x^2+5x-8 .



```

101 % --- Executes on button press in Calcula.
102 function Calcula_Callback(hObject, eventdata, handles)
103 % hObject    handle to Calcula (see GCBO)
104 % eventdata  reserved - to be defined in a future version of MATLAB
105 % handles    structure with handles and user data (see GUIDATA)
106 % ======================================================================
107 ecuacion = eval(get(handles.la_ecuacion, 'String'));
108
109 derivada1 = diff (ecuacion);
110 derivada2 = char (derivada1);
111
112 integral1 = int (ecuacion);
113 integral2 = char (integral1);
114
115 Fourier1 = fourier (ecuacion);
116 Fourier2 = char (Fourier1);
117
118 set (handles.resultado_derivada, 'String', derivada2 );
119 set (handles.resultado_integral, 'String', integral2 );
120 set (handles.resultado_Fourier, 'String', Fourier2 );
121
122

```

Figura 7.17
Código del botón
Calcula.

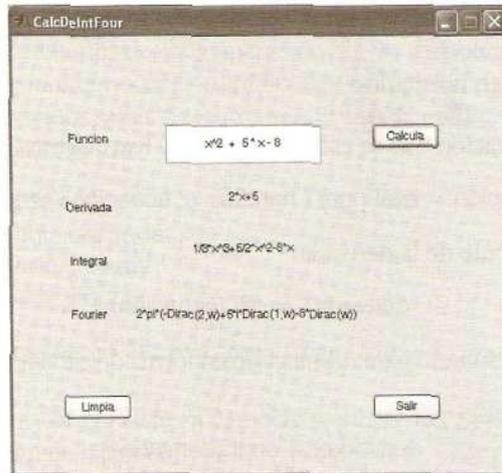


Figura 7.18 Prueba de la aplicación CalcDeIntFour.

EJEMPLO 7.3 Conversor de temperaturas

Nuestro siguiente ejemplo es un conversor de temperaturas que nos permite convertir entre las escalas de temperatura Celsius-Kelvin-Fahrenheit. Para empezar ejecutamos Guide como se ha visto y colocamos cuatro `StaticText` como se muestra en la figura 7.19, dándoles el tamaño indicado en la figura.

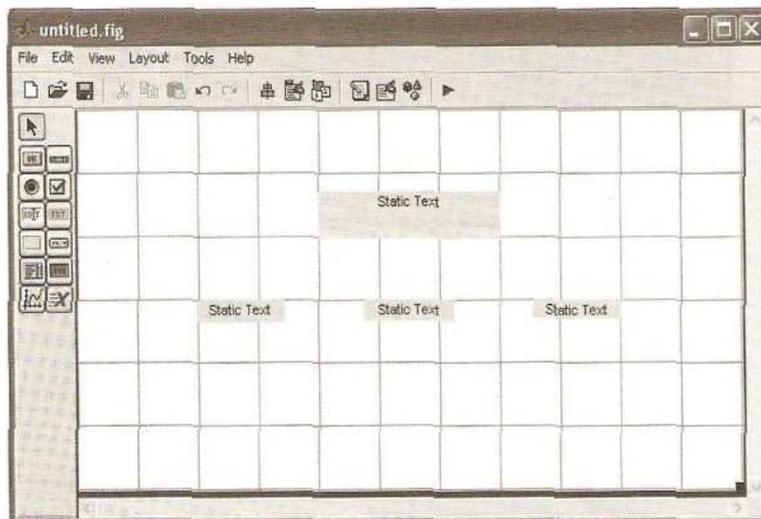


Figura 7.19 Posición y tamaño de los `StaticText`.

En el caso de cada una de ellas modificamos la propiedad String con el Property Inspector en el siguiente orden:

Calculador de Temperaturas
Celsius
Kelvin
Fahrenheit

En la figura 7.20 se puede apreciar el estado actual de la GUI, mientras que la figura 7.21 muestra la ventana del Inspector de Propiedades con el cambio del último Static Text a Fahrenheit

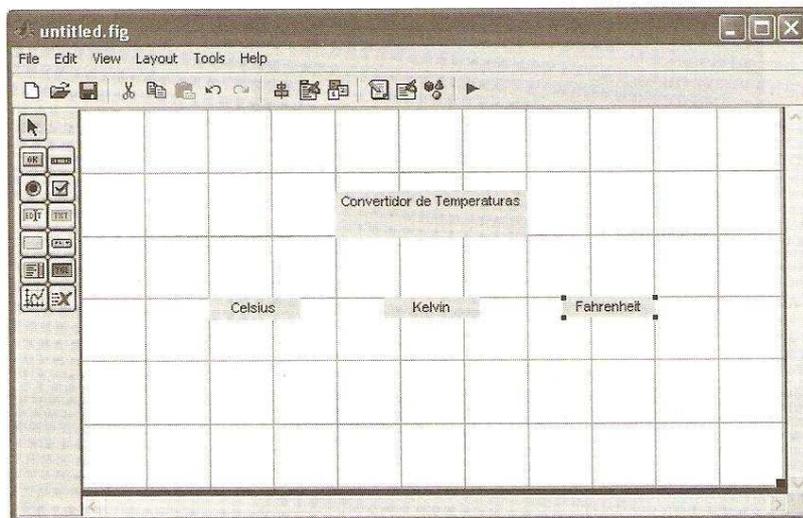


Figura 7.20 Estado actual de la GUI.

Debajo de las etiquetas Celsius, Kelvin, Fahrenheit agregamos tres Edit Text y cuatro Push Button y modificamos la propiedad String para que los Edit Text queden en blanco, quitando todo texto de la propiedad String, y en los Push Button cambiamos la propiedad String a Calcula para los tres primeros y Salir para el último, para dejar ahora la interfase como se muestra en las figura 7.22. Al mismo tiempo cambiamos la propiedad Tag de los Push Button a Boton_Celsius, Boton_Kelvin y Boton_Fahrenheit, respectivamente, como se muestra en la figura 7.23 para el botón Celsius. Para el botón Salir le cambiamos su Tag a Boton_Salir.

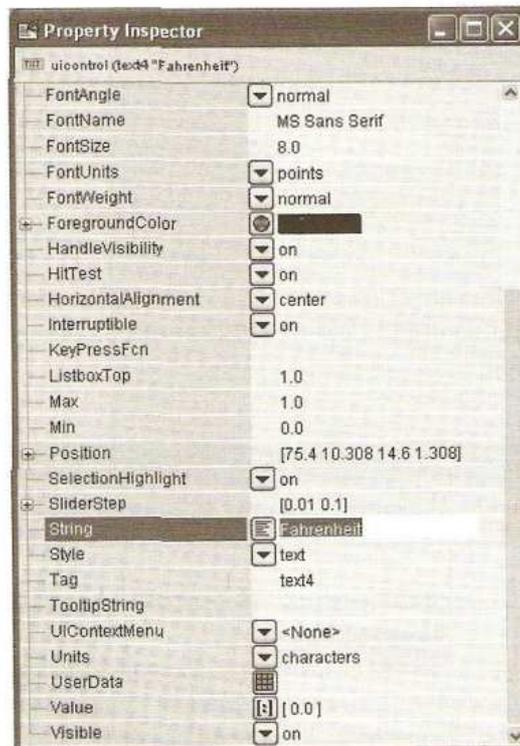


Figura 7.21
Cambio del nombre
en el Inspector de
Propiedades.

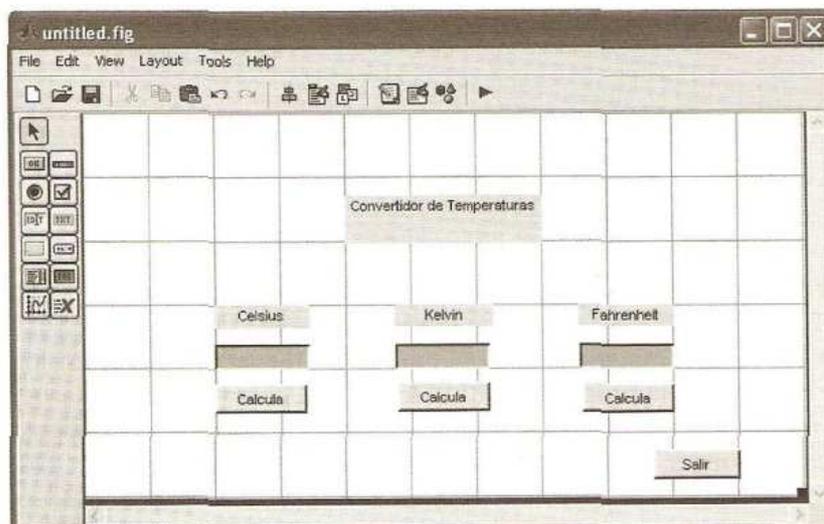


Figura 7.22 La interfase con los nuevos controles.