

Informática II Práctica No. 5.1

Introducción al desarrollo de GUIs con *Qt Developer*

Objetivos

- Realizar una introducción al IDE Qt Creator para la construcción de interfaces gráficas de usuario.
- Construir una interfaz gráfica simple empleando los conceptos de signal y slot.

1. Creando una Interfaz Gráfica de Usuario (GUI)

Qué es un Widget? Un Widget es un componente visual, por ejemplo, un botón, un cuadro de texto, los ítems de un menú, o cualquier otro elemento que hace parte de la interfaz gráfica de usuario.

Para crear un proyecto con interfaz gráfica, hacemos click en File->New File or Project->Qt Widget Project->Qt Gui Application y navegamos a través de los cuadros de diálogo que se presentan a continuación. Nótese, dentro de la carpeta Forms del proyecto creado, el archivo mainwindow.ui. Los archivos con extensión *.ui representan la especificación de la interfaz de usuario y pueden ser editados gráficamente. Al hacer doble click sobre mainwindow.ui se habilita el modo de diseño (la pestaña Design) y podemos entonces comenzar a agregar widgets a nuestra interfaz gráfica.

Para agregar un Widget, estos se arrastran desde la parte izquierda y se descargan en la parte central (ventana). Agregue un dial, un lcdNumber y un pushButton, como se muestra en la Figura 1. Compile y ejecute la aplicación.

1.1 Signals y Slots

Las señales (signals) y los slots se usan para la comunicación entre objetos. Una señal se emite cuando un evento particular ocurre (por ejemplo, si se presiona un botón con el mouse o se arrastra una barra de desplazamiento). Los Widgets de Qt tienen muchas señales predefinidas y el usuario también puede definir sus propias señales.

Un slot es una función que es llamada en respuesta a una señal particular. Los slots son funciones miembro comunes y corrientes, y los Widgets de Qt también tienen muchos definidos por defecto.

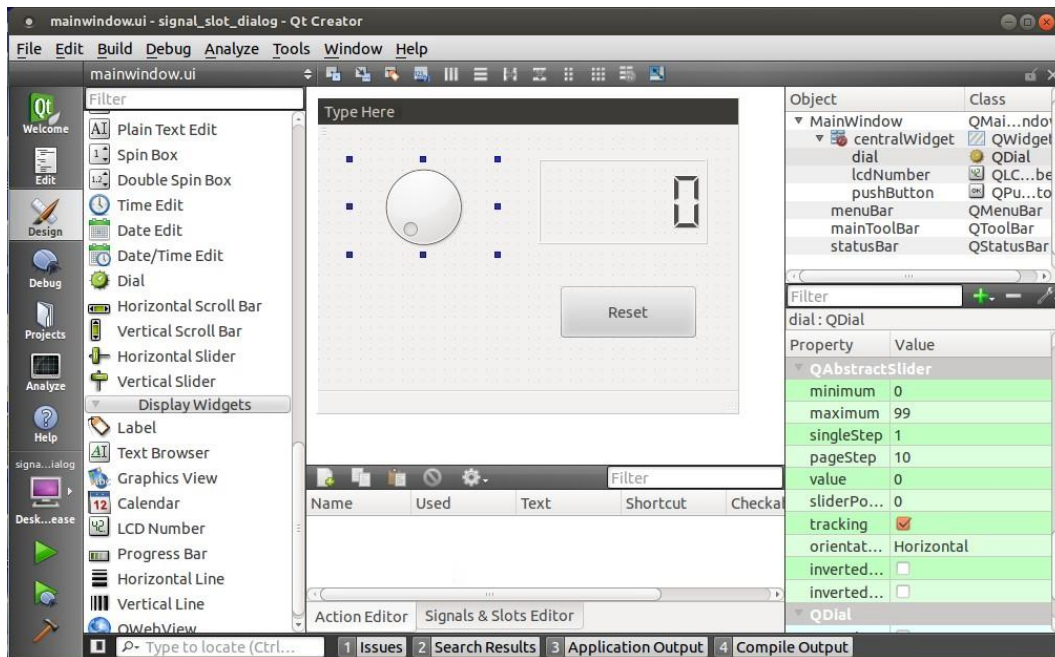


Figura 1. Ejemplo de interfaz gráfica usando widgets

El mecanismo de signals y slots garantiza que si se conecta una señal a un slot, la función representada por el slot será llamada con los parámetros de la señal en el momento indicado.

La conexión entre signals y slots se puede realizar de manera gráfica o mediante código. Para proceder gráficamente, realice la conexión de la señal `valueChanged` del dial con el slot `display` del `lcdNumber`, como lo ilustra la Figura 2.

Luego, asigne el nombre `Reset` para el `pushButton`, haga click derecho sobre este y escoja la opción `Go to slot...`, luego seleccione la señal `clicked()` tal como se muestra en la Figura 3.

Dentro del slot generado en el archivo `mainwindow.cpp` para la señal `clicked()` inserte la línea:

```
ui->lcdNumber->display(0);
```

Compile y ejecute la aplicación y observe que sucede al mover el dial y presionar el botón.

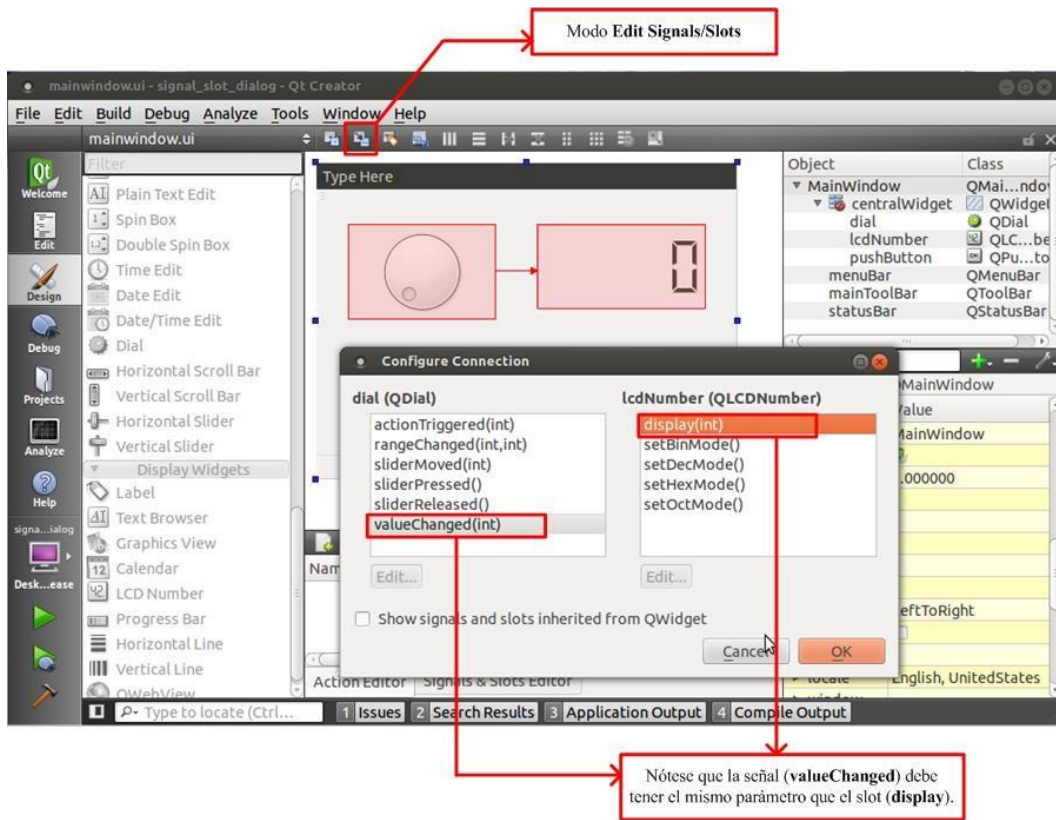


Figura 2. Conexión entre signals y slot usando el asistente.

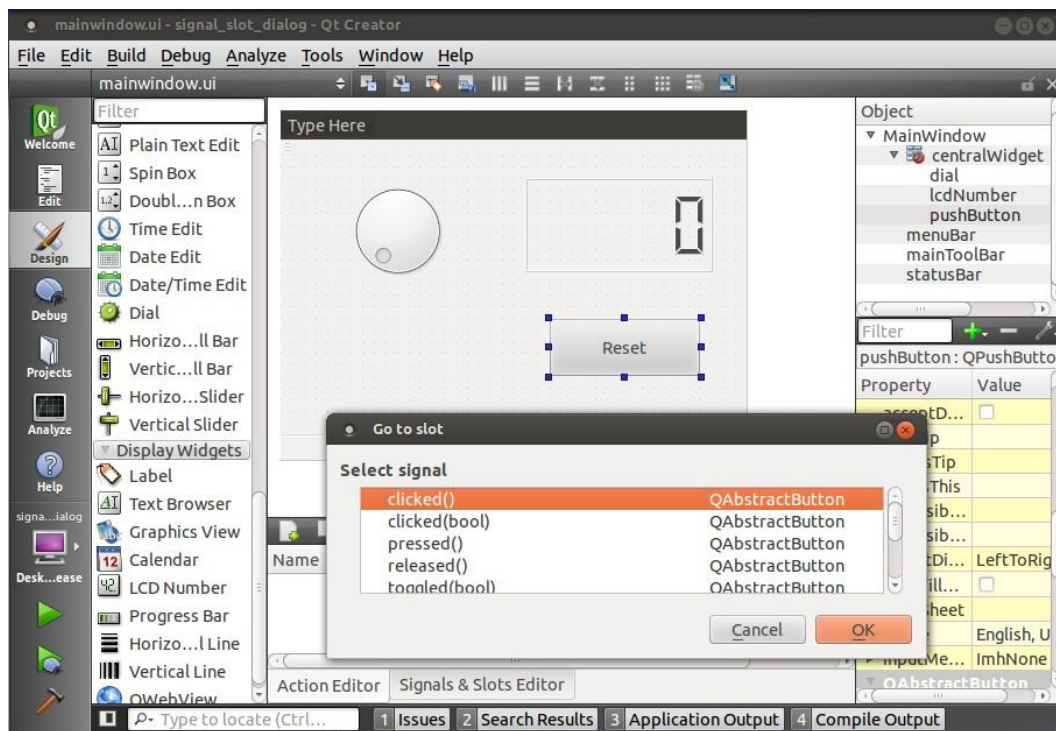


Figura 3. Administración de slots del widget PushButton.

1.2 Conexión mediante código de signals y slots

Elimine la conexión realizada anteriormente de manera gráfica y agregue las siguientes líneas al constructor de la clase MainWindow:

```
connect(ui->dial, SIGNAL(valueChanged(int)), ui->lcdNumber, SLOT(display(int)));
```

Compile y ejecute de nuevo la aplicación.

1.3 Agregando un diálogo a la interfaz gráfica

Una Ventana provee el espacio en pantalla sobre el cual se construye la interfaz gráfica principal, a menudo las ventanas tienen barras de menú y de estado. En su parte central se puede añadir cualquier tipo de Widget (QWidget). Las ventanas se construyen derivando la clase QMainWindow. Los Diálogos, son ventanas auxiliares que muestran información adicional o le presentan al usuario diversas preguntas y/o elecciones. Los diálogos se construyen derivando la clase QDialog y agregando a este los Widget necesarios.

Para agregar un diálogo a nuestra aplicación, hacemos click derecho en la carpeta Forms y seleccionamos Add New->Qt->Qt Designer Form Class->Dialog Without Buttons.

Al hacer doble click sobre el archivo generado dialog.ui, se cambia al modo diseño. Adicione una etiqueta label con el nombre calendario y un Widget tipo calendario al diálogo. El formulario debe lucir como se muestra en la Figura 4.

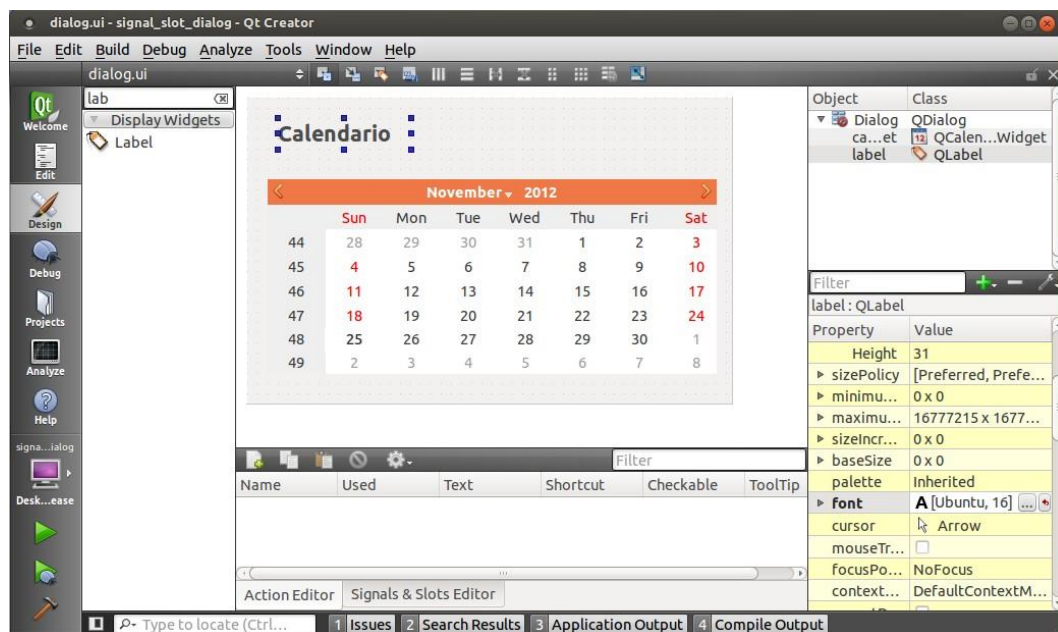


Figura 4. Aplicación con varias ventanas.

1.4 Creando un menú en la ventana principal

De regreso a la vista de diseño para la ventana principal (doble click sobre mainwindow.ui).

Localice la barra de menú de la ventana y haga click en la etiqueta Type Here y reemplace este nombre con File o Archivo. Cree en la pestaña del nuevo menú File, la opción Calendar o Calendario. El formulario debe lucir como se muestra en la Figura 5.

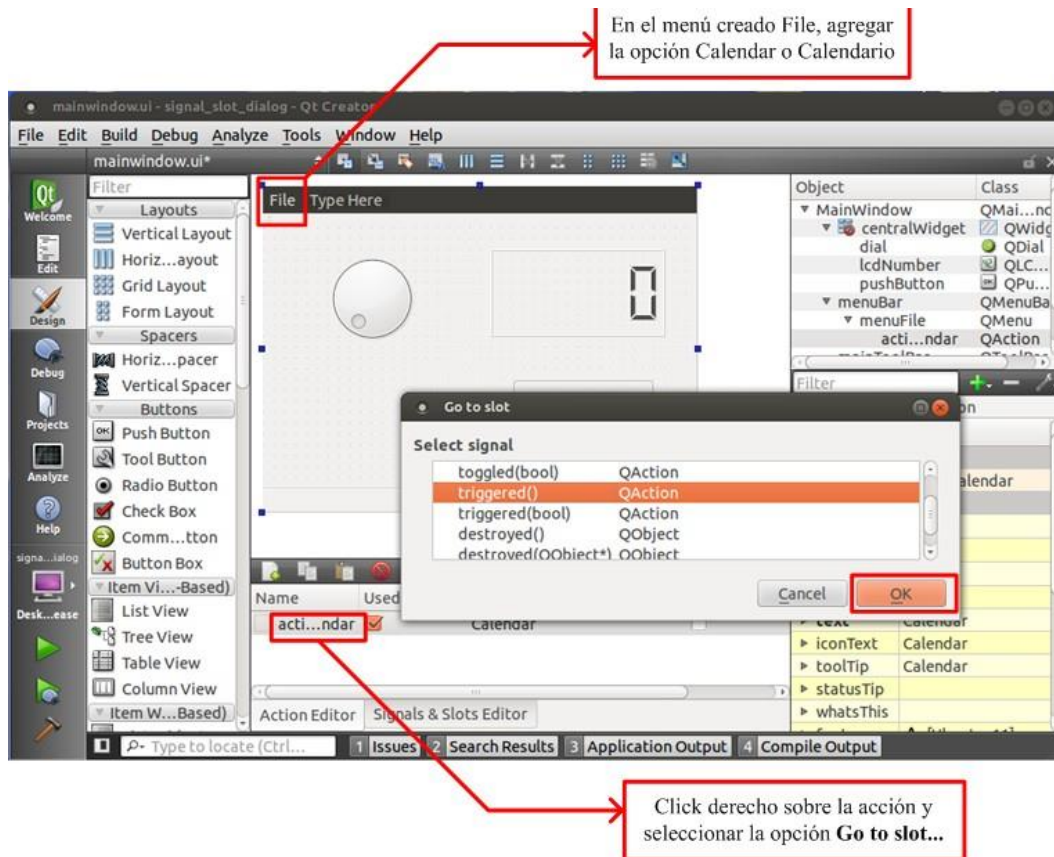


Figura 5. Agregando un menú.

Ingresa las siguientes líneas de código en el slot generado para la señal triggered() de la acción Calendario (recuerde incluir el archivo dialog.h en mainwindow.cpp):

```
Dialog *MyDialog = new Dialog(); MyDialog->show();
```

Complete el ejemplo agregando una opción al menú File que permita cerrar la ventana principal.

2. Ejercicios

- Elabore un editor de texto plano, con un menú que contenga una opción para limpiar el texto escrito.
- Elabore una calculadora, con una interfaz gráfica similar a la que se muestra en la Figura 6.

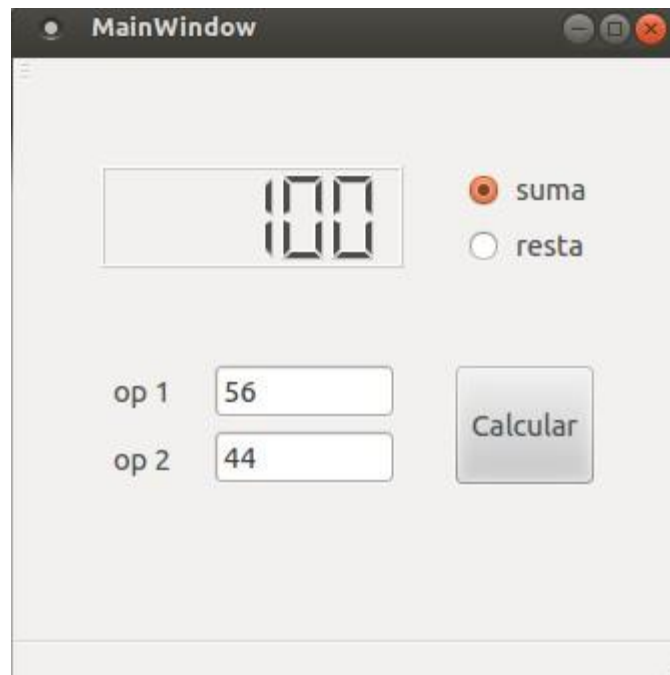


Figura 6. Aplicación: Calculadora.

Esta guía fue desarrollada por Felipe Lennis y modificada por David Jaramillo y Augusto Salazar