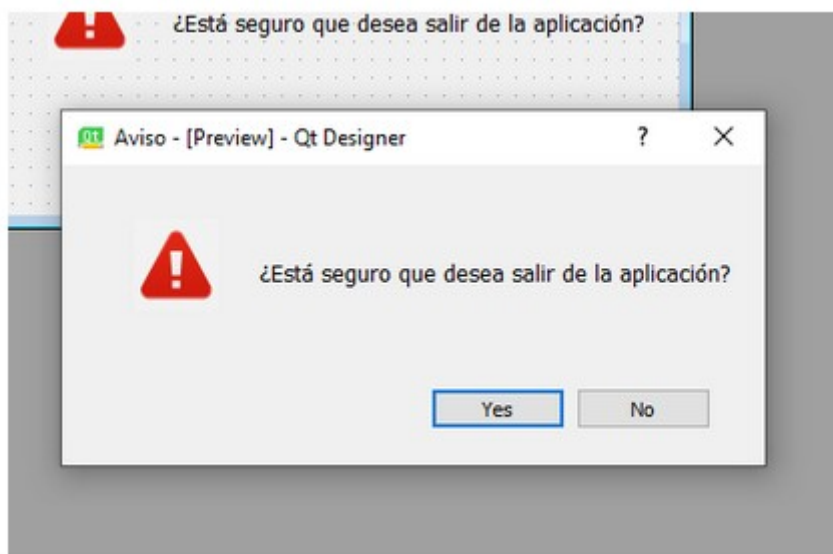


## Ventanas modales

Las **ventanas modales** son ventanas o cuadros de diálogo que aparecen sobre la página, bloqueando todas las funciones para concentrar el foco en una acción particular. Esta es su característica diferenciadora, le piden al usuario realizar una acción. Se suele identificar cuando el fondo de la ventana primaria se oscurece. En caso contrario se denominan **no-modales**.

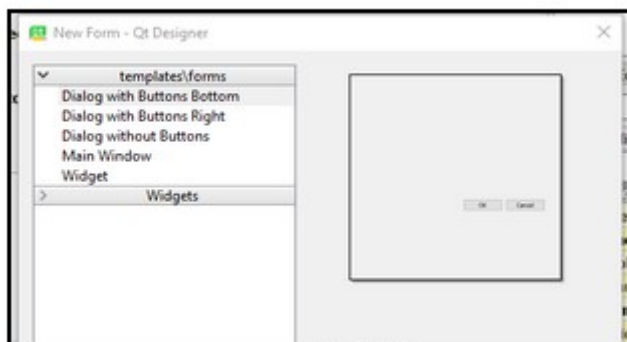
Ventanas del tipo: **pop up**, **pop over** y **light box**, son derivadas de estas.

Como ejemplo, vamos a generar una ventana modal que nos pregunte previamente si queremos salir de la misma.

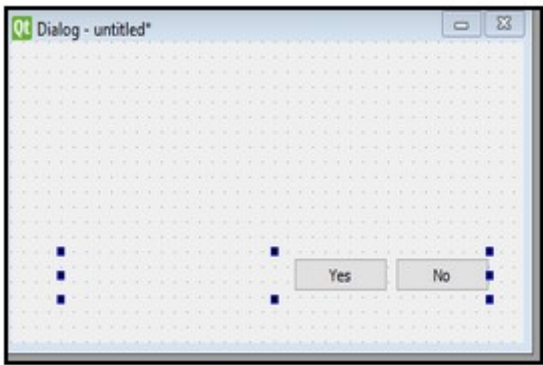


Los pasos necesarios serán:

Lanzar una ventana de **diálogo**



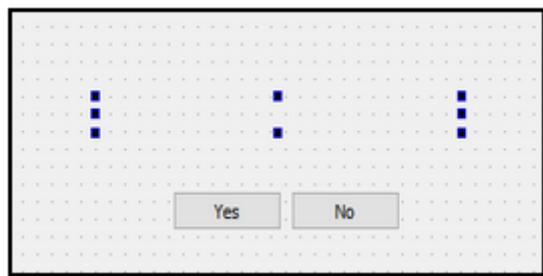
El widget de los botones es un **button dialogbox** que permite contener una serie de botones



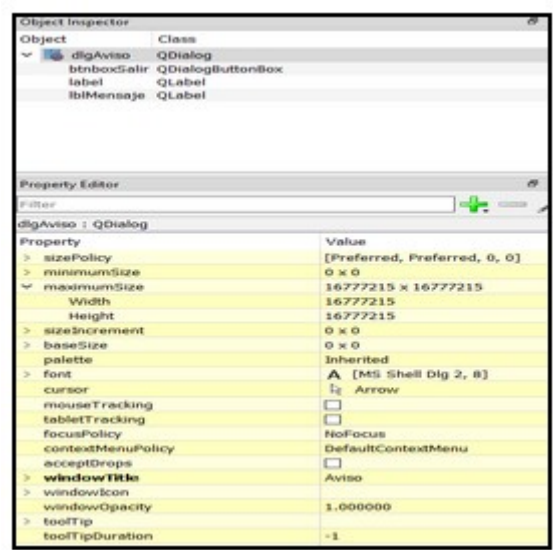
Algunas propiedades del **button Dialogbox**:

inputMethodHints	ImhNone
QDialogButtonBox	
orientation	Horizontal
standardButtons	Yes No
NoButton	<input type="checkbox"/>
Ok	<input type="checkbox"/>
Save	<input type="checkbox"/>
SaveAll	<input type="checkbox"/>
Open	<input type="checkbox"/>
Yes	<input checked="" type="checkbox"/>
YesToAll	<input type="checkbox"/>
No	<input checked="" type="checkbox"/>
NoToAll	<input type="checkbox"/>
Abort	<input type="checkbox"/>
Retry	<input type="checkbox"/>
Ignore	<input type="checkbox"/>
Close	<input type="checkbox"/>
Cancel	<input type="checkbox"/>
Discard	<input type="checkbox"/>
Help	<input type="checkbox"/>
Apply	<input type="checkbox"/>
Reset	<input type="checkbox"/>
RestoreDefaults	<input type="checkbox"/>
centerButtons	<input checked="" type="checkbox"/>

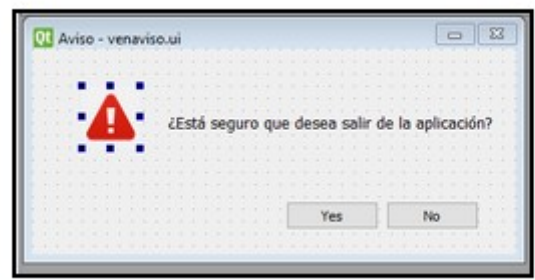
El label que contendrá el texto:



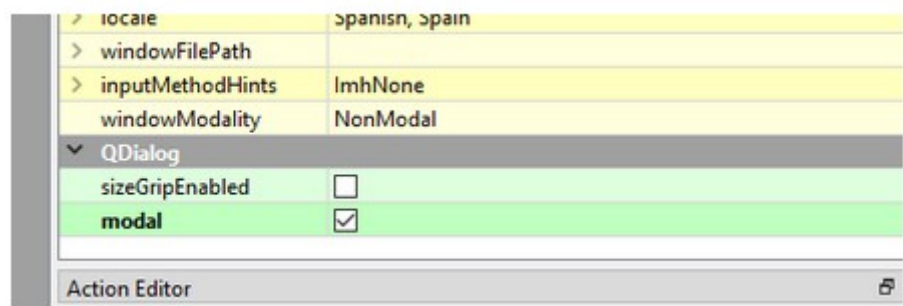
Propiedades de la ventana de dialogo:



La imagen debe de ir en un label o etiqueta:



Al final de las propiedades del editor, hay que marcar **modal**

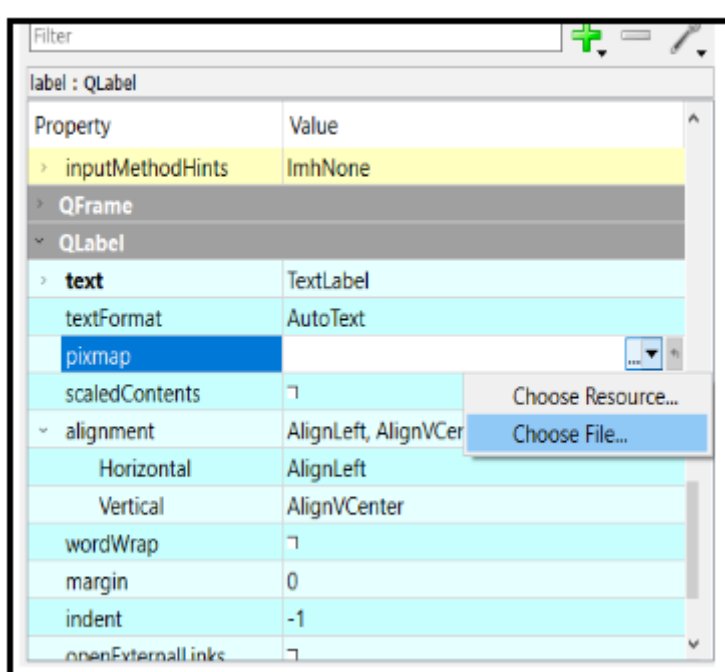


La inserción de la imagen de *warning* en la **ventana modal** requiere de una serie de pasos. En primer lugar,

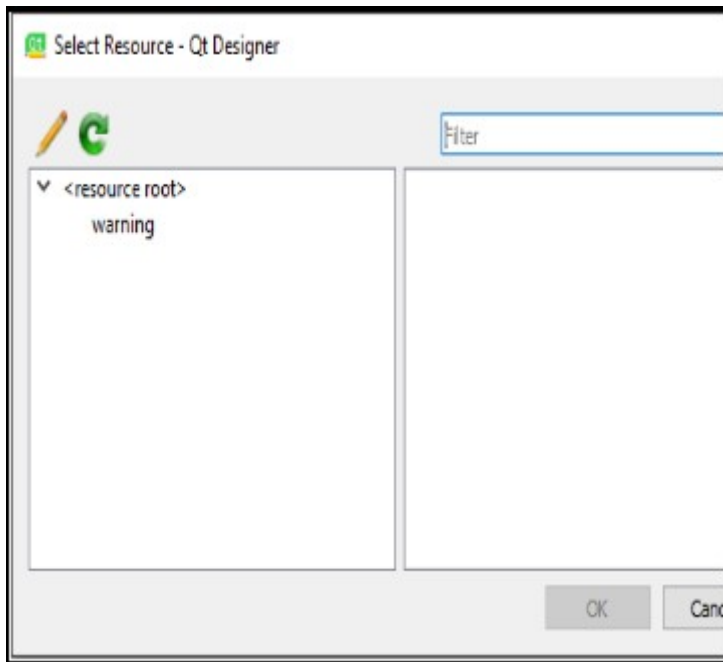
1. Creamos una carpeta llamada **img** en el directorio de la aplicación
2. Descargamos una imagen similar a la del ejemplo y escalamos su tamaño a 36x36 pixeles guardándola en el directorio anterior, aunque se puede obviar este paso, ya que Qt-Designer escala las imágenes, pero haciéndolo descargamos el peso de la aplicación en su distribución.

En segundo lugar, desde *Qt-Designer* seguimos las indicaciones de las imágenes que se muestran:

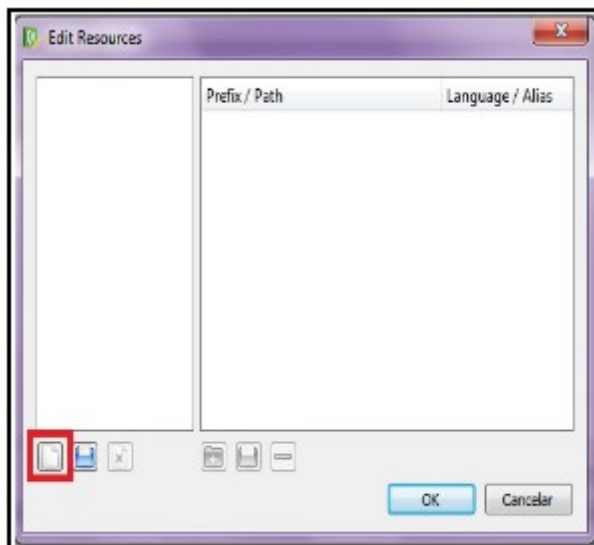
**Pixmap** en Label, donde configuraremos la carga de la imagen con **Choose File**



Luego en **Choose Resource...**

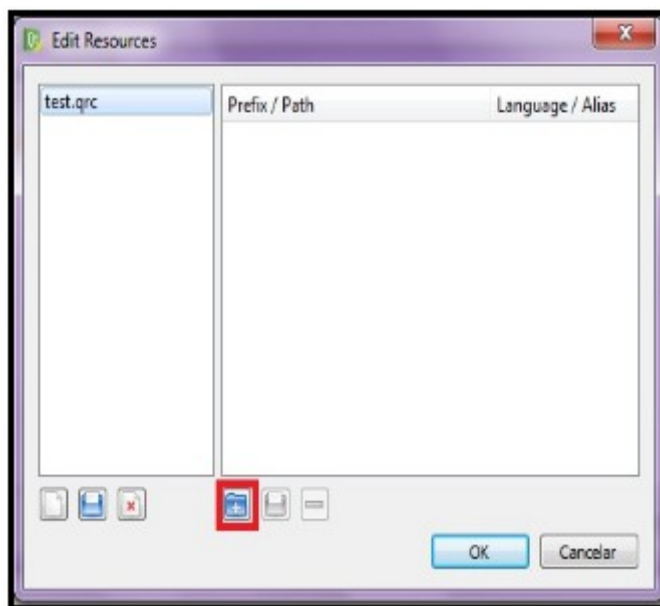


donde pulsamos en el **lápiz** con lo que se nos muestra la imagen siguiente:

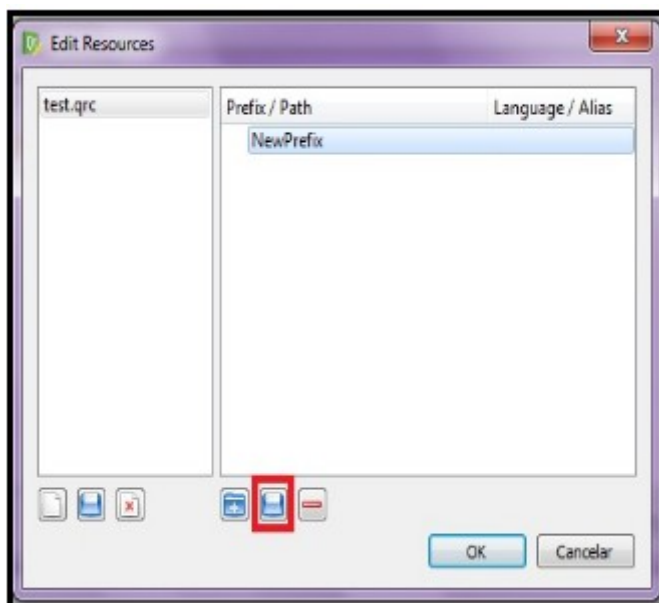


Donde pulsamos **new resource file** y dentro de **img** le ponemos un nombre

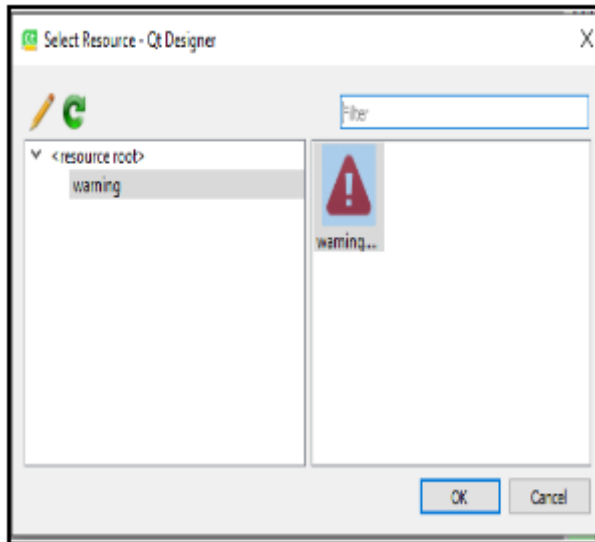
Añadimos el **prefix**, mecanismo para hacer referencia de **la imagen**



Subimos la imagen descargada de la red y pulsamos OK



Resultado final que se debe obtener. Vemos el fichero de la imagen y el **src** al que se asocia:



**Importante**, debemos marcar **scaledContents** para que la imagen se adapte al tamaño del label.

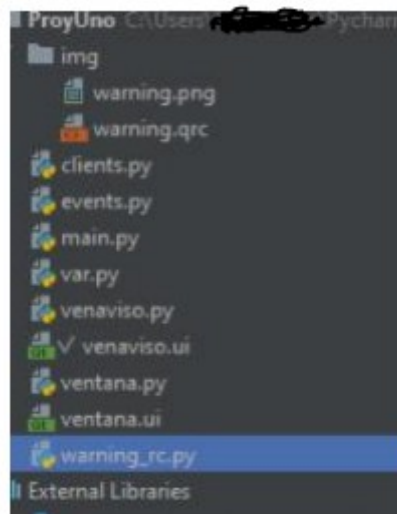
Ahora pasamos al código. En este caso, lo que pretendemos es que cuando pulsemos Salir, la aplicación nos pregunte antes si deseamos hacerlo, y si confirmamos salimos de la aplicación y, en caso contrario, volvemos a la situación inicial. Debemos compilar la **ventanasalir.ui** a **.py**

En primer lugar tenemos que convertir el **recurso .qrc en .py**. Con PyQt tenemos que convertir todo fichero que forme parte del proyecto en un fichero **python**. Para la conversión a fichero python:

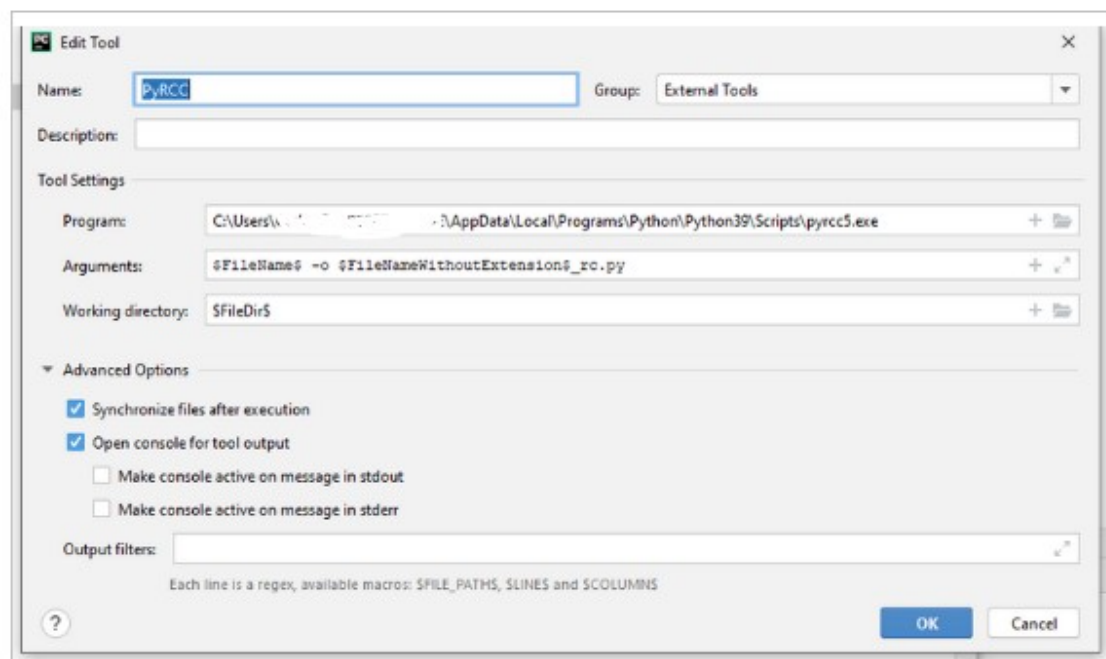
```
\ProyUno\ing>pyrcc5 warning.qrc -o warning_rc.py  
  
\ProyUno\ing>
```

**Muy importante**, el fichero generado debe encontrarse **en la misma ruta** que el resto de los ficheros Python.

Y el fichero Python creado:



Si se quiere automatizar la compilación .py del fichero como una **External Tools**, la configuración de **File -> Settings** sería:



Quedaría incluir el código y conectar los botones de la ventana de diálogo al mismo.

```
class DialogSalir(QtWidgets.QDialog):
    def __init__(self):
        super(DialogSalir, self).__init__()
        var.avisosalir = Ui_dlgSalir()
        var.avisosalir.setupUi(self)
        var.avisosalir.btnBoxSalir.button(QtWidgets.QDialogButtonBox.Yes).clicked.connect(events.Eventos.Salir)
        #var.avisosalir.btnBoxSalir.button(QtWidgets.QDialogButtonBox.No).clicked.connect(events.Eventos.Salir)
        #no es necesario no quiero que haga nada
```



```
def Salir(event):
    """
    Módulo para cerrar el programa
    :return:
    """
    try:
        var.avisosalir.show()
        if var.avisosalir.exec_():
            sys.exit()
        else:
            var.avisosalir.hide()
            event.ignore() #necesario para que ignore X de la ventana
    except Exception as error:
        print('Error %s' % str(error))
```

En el fichero **eventos.py** modificamos el método **Salir** adaptándolo para que llame a la ventana **dialog** tal como se muestra.

En el fichero **main.py** se han hecho varias modificaciones.

- Incluimos la clase **DialogSalir** para instanciar la ventana diseñada, importando además el fichero de la misma **venaviso.py** en la clase principal.
- Configuramos **QAction** para que la ventana **dialog** de Salir sea llamada cuando **pulsamos el icono X** de la principal. Son las líneas:

```
var.dlgcalendar = DialogCalendar()
QtWidgets.QAction(self).triggered.connect(self.close) #debe cargarse al principio para estar a la escucha

def closeEvent(self, event):
    events.Eventos.Salir(event)
```

El evento **QAction** ya está configurado por defecto en la clase **QWidget**. Sin embargo, tal como se ha hecho, se puede sobrescribir el método manejador del evento, por ejemplo, nuestra ventana extiende la clase **QWidget**, esta utiliza el método **closeEvent(self, event)** para responder al evento que se produce cuando el usuario hace clic sobre el botón de cerrar (X), la acción es cerrar la ventana, podemos sobrescribir este método para que tenga un comportamiento diferente, como es el caso, preguntar al usuario si realmente desea cerrar la aplicación.