

# Mi primera aplicación gráfica en Python

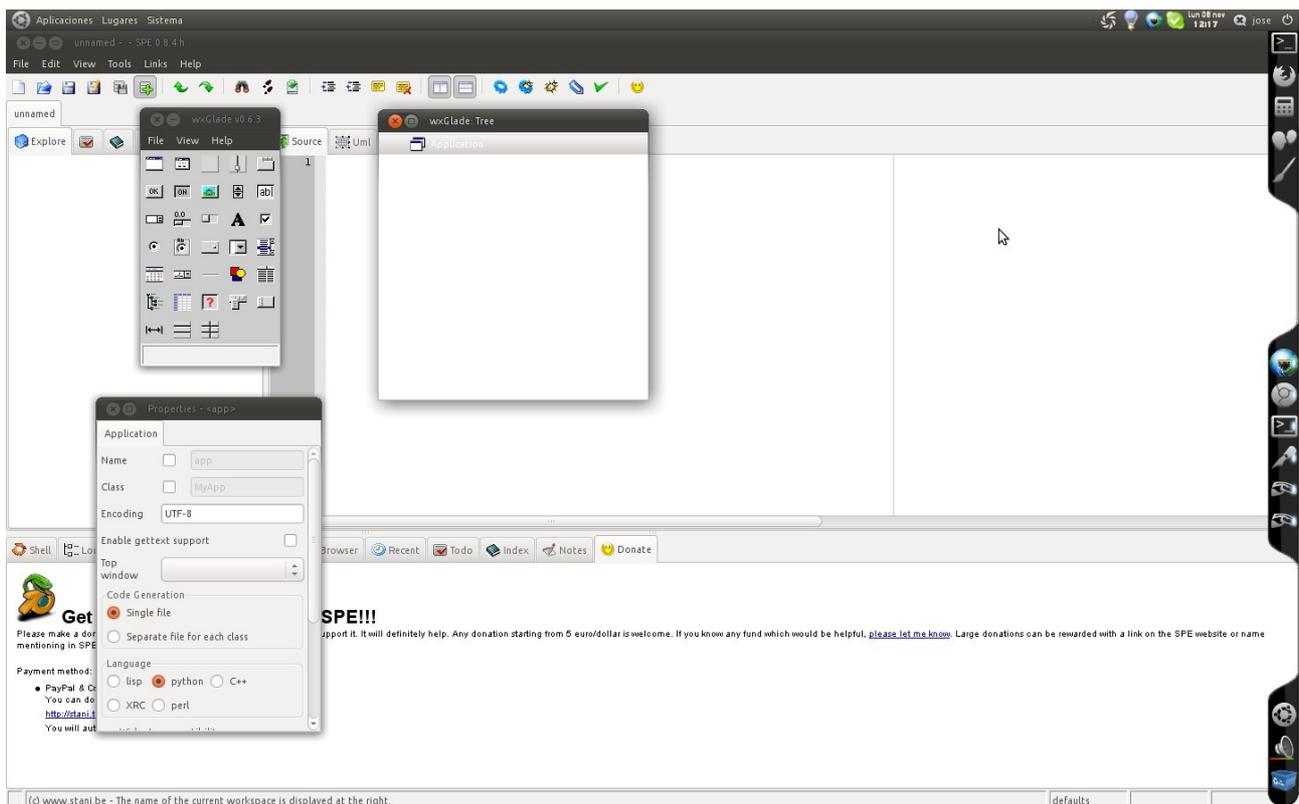
Bueno, realmente ésta no es mi primera aplicación gráfica en Python, pero de alguna manera tenía que llamar a este minitutorial (no lo considero tampoco un tutorial completo al 100% pues he ido un poco rápido en su escritura) pero si que espero que sea de utilidad para todos aquellos que quieran empezar con el desarrollo grafico.

En este breve minitutorial veremos como crear una pequeña aplicación gráfica para dar de alta repositorios PPA (MiPPA) y puede servir de base para desarrollos mayores. Como IDE de desarrollo utilizo SPE (Stani's Python Editor) que puedes instalar con:

**\$ sudo apt-get install spe**

Pero seguramente sea facilmente extrapolar lo aquí comentado a cualquier otro IDE. Bien, Empezamos.

Arranca el entorno grafico de SPE y directamente abrimos ya el asistente grafico para crear el esqueleto de la interface: **Tools – Design a gui with wxGlade...**



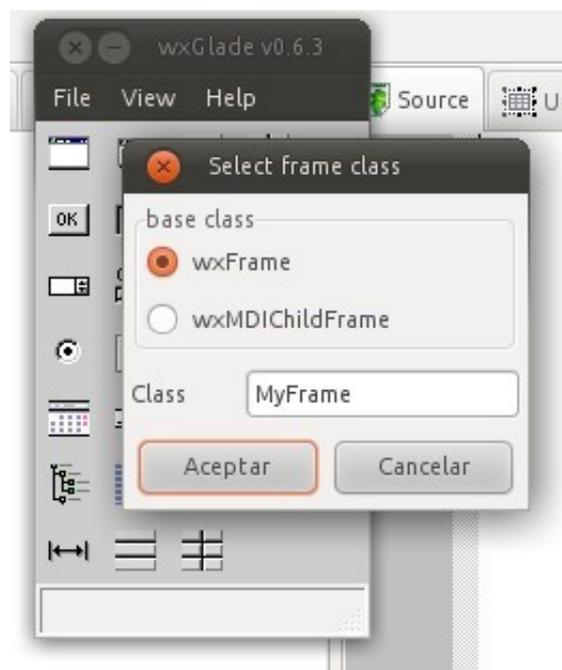
Si te fijas se abren varias ventanas. WxGlade es una aplicación que nos permitira construir visualmente la interface grafica de nuestra aplicación (se basa en la tecnica de arrastrar y soltar componente de la paleta de control) y una vez que tengamos el interface definido es capaz de autogenerar el codigo fuente correspondiente para el manejo de esa interface. De las 3 ventanas nuevas que se abren distinguimos:

- La paleta de controles visuales. Muestra los diferentes elementos que podemos usar en la creacion de la interface grafica: botones, cajas de texto, paneles, etc...



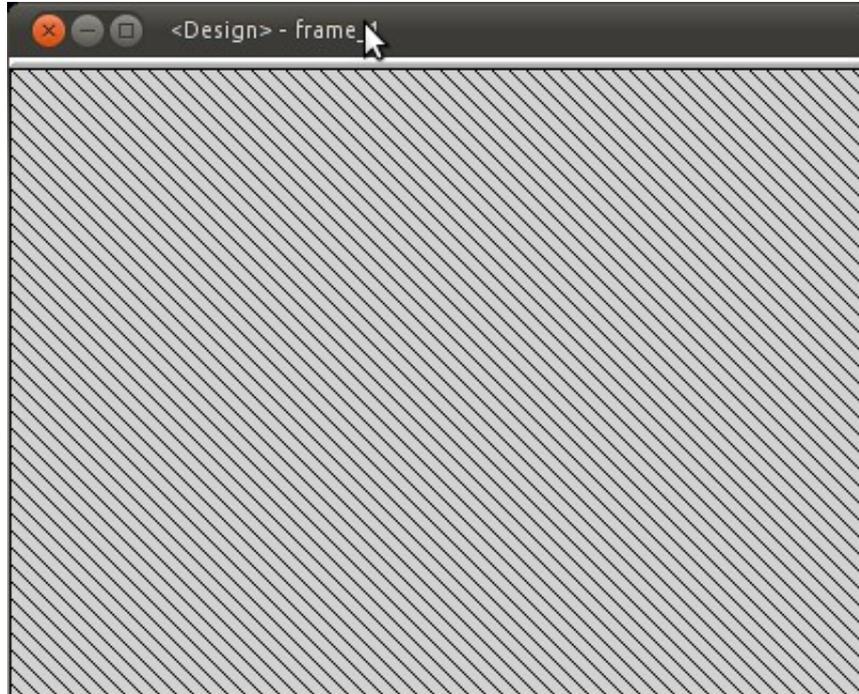
Tambien esta la ventana de propiedades. Que son los valores que podemos cambiar de un control cuando lo tenemos seleccionado. Y por ultimo la ventana de arbol, que nos muestra la estructura jerarquica de controles visualmente de la pantalla o formulario que actualmente estemos diseñando. Desde esta ventana en arbol podremos seleccionar facilmente un control que tengamos añadido para editar sus propiedades o borrarlo por ejemplo.

La idea basicamente sera ir arrastrando los controles que queramos usar a la ventana de diseño para ir construyendo nuestra ventana. Empecemos con el primer control que aparece en la esquina superior izquierda, es el Frame. Al hacer click sobre el creamos lo que sera la ventana principal de nuestra aplicación.

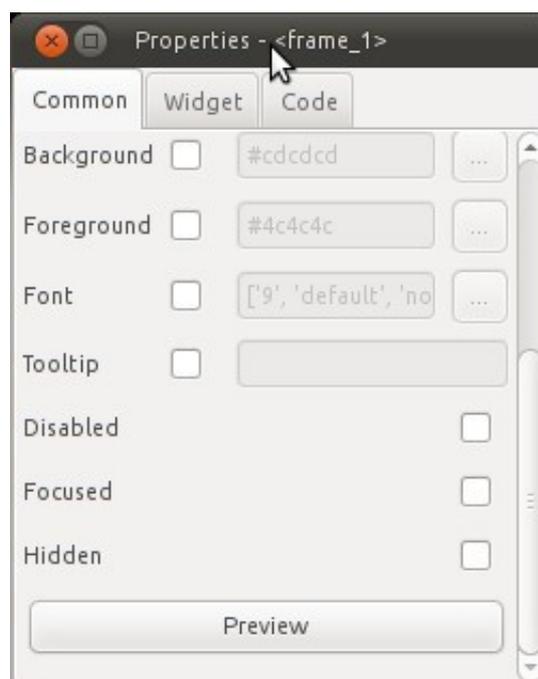


Selecciona “wxFrame” y si lo deseas indica ya un nombre para la clase. Con el nombre de la clase podremos acceder luego a las propiedades del elemento desde el código fuente, así que es recomendable utilizar nombres que luego podamos identificar fácilmente.

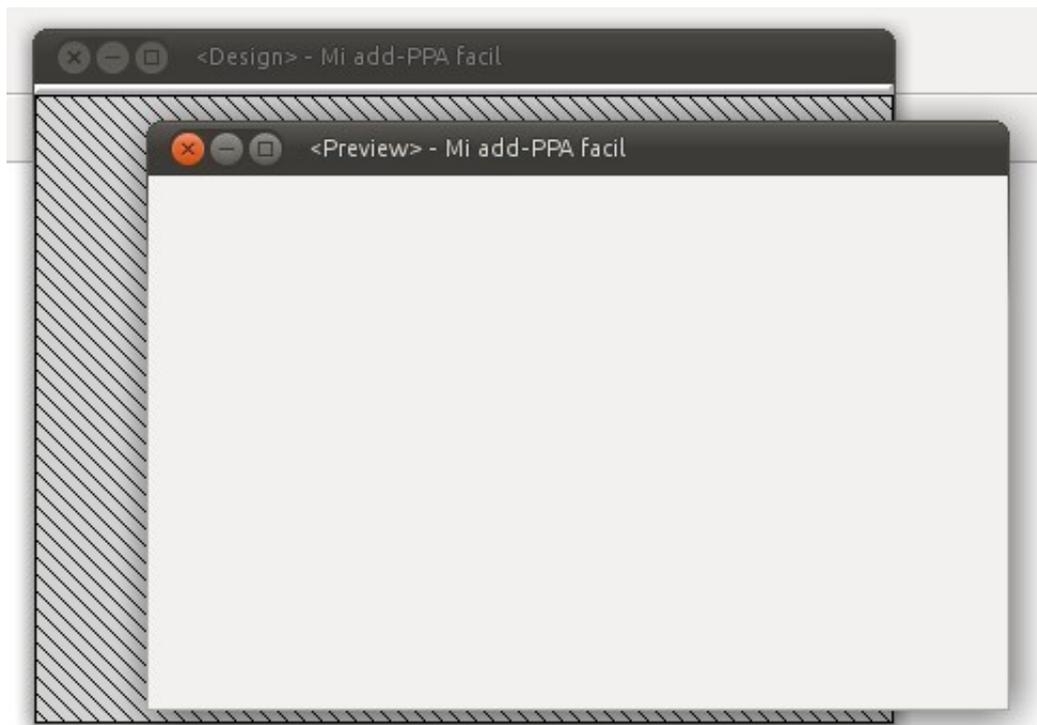
Bien al añadir el frame se crea la vista de diseño de la ventana:



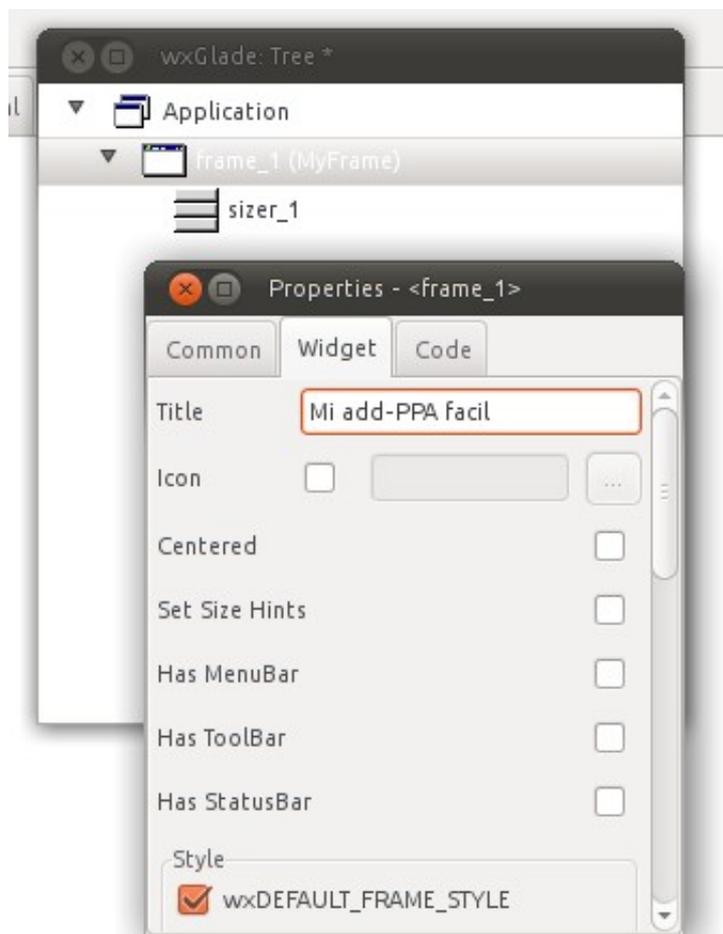
Es a esta ventana a la que podemos ir arrastrando componentes de la paleta de control hasta ir creando la interfaz como más nos apetezca. En cualquier instante podemos también ver cómo va quedando la ventana (modo Preview) si seleccionamos este Frame y en la vista de propiedades de la misma pulsamos sobre el botón Preview:



Así podemos ir viendo su aspecto actual:



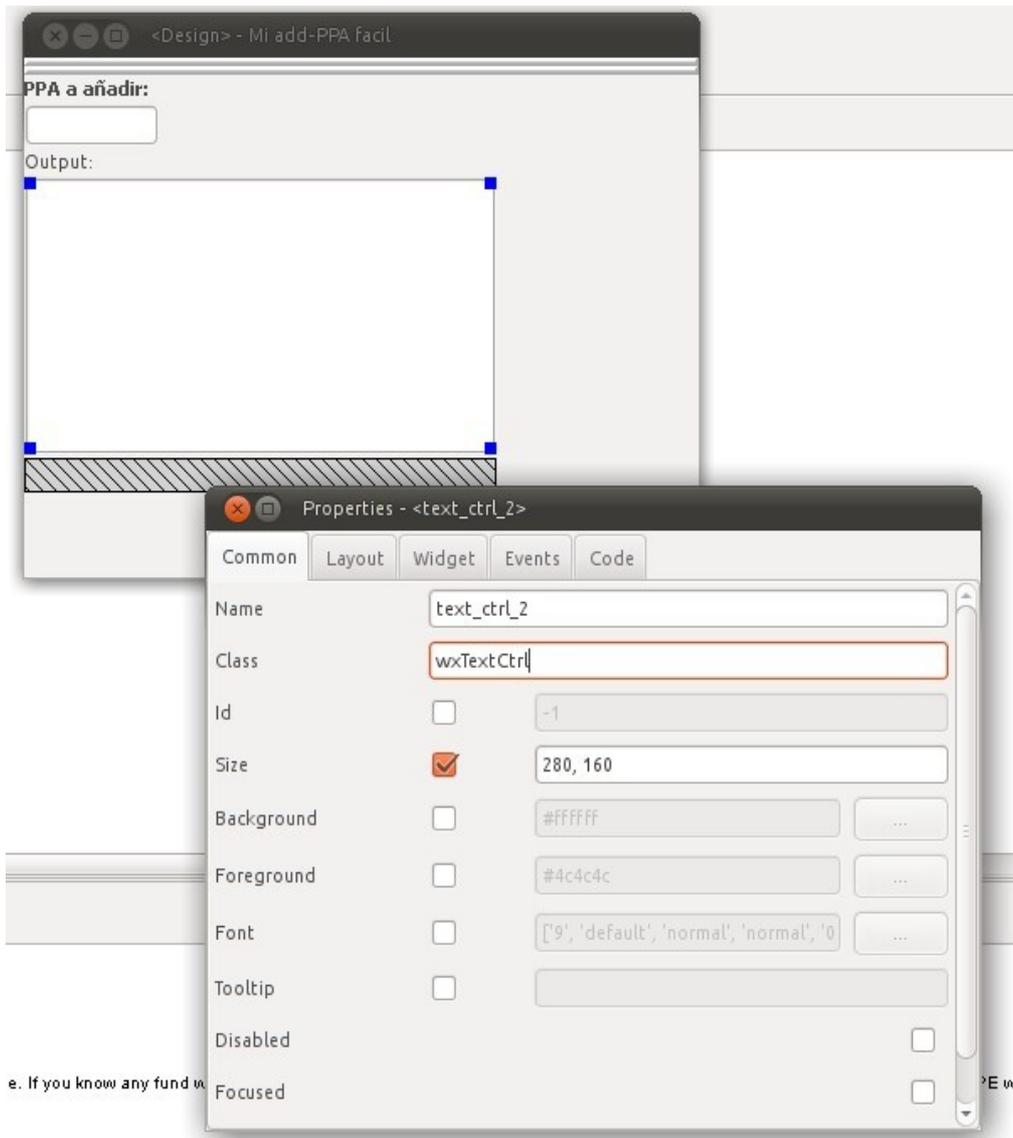
Si te fijas, el título de la ventana ha cambiado, Eso es precisamente porque ya he editado una de sus propiedades "title", que permite indicar el título de la ventana.



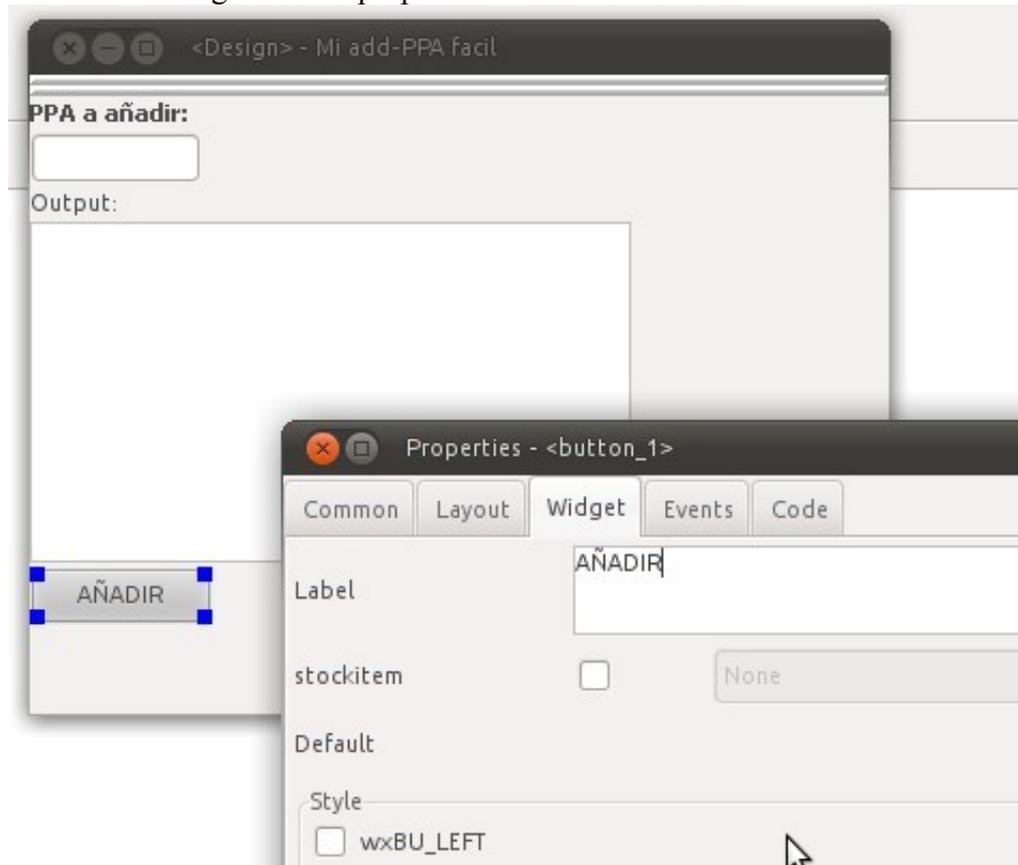
Pues como comentaba, tienes que ir “dando forma” a la ventana, añadiendo los elementos que quieras usar y ajustando individualmente sus propiedades. Para que los elementos queden bien alineados, es conveniente que utilices un control llamado “grid sizer” y que crea una especie de rejilla (o tabla) sobre los que puedes situar el resto de elementos:



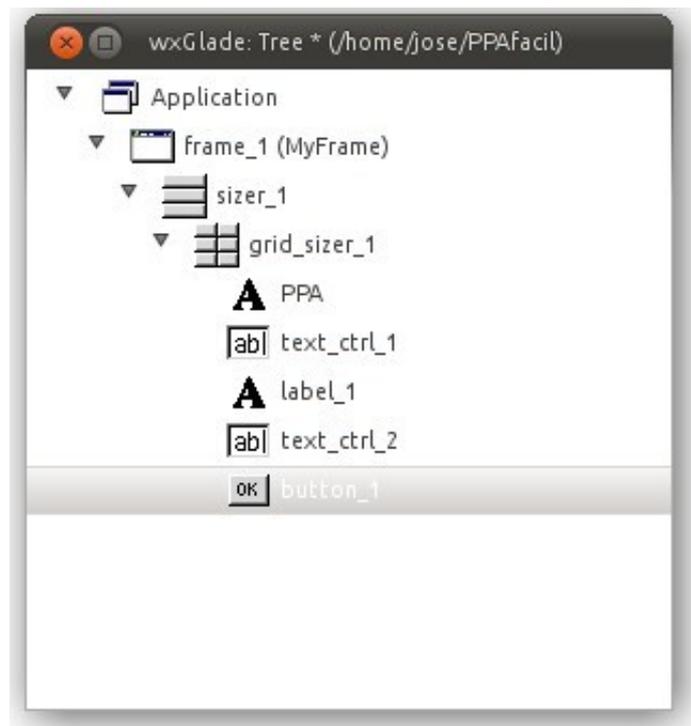
Es hora de ir metiendo elementos, yo he añadido una etiqueta (Label), una caja de texto y un boton:



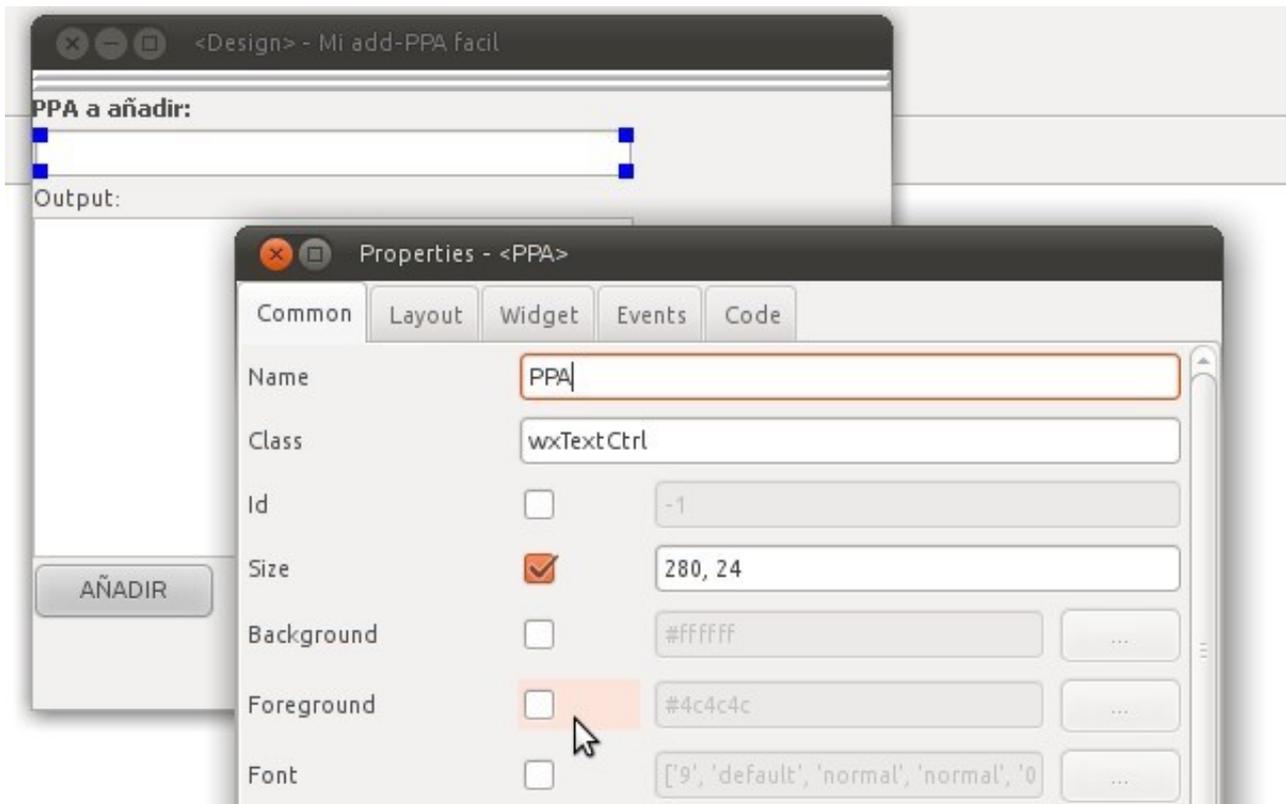
Ejemplo, Añadiendo el boton configuro como propiedad "Label" el nombre:



En cualquier momento podemos ver como va quedando la jerarquia o seleccionar elementos desde la vista de arbol:

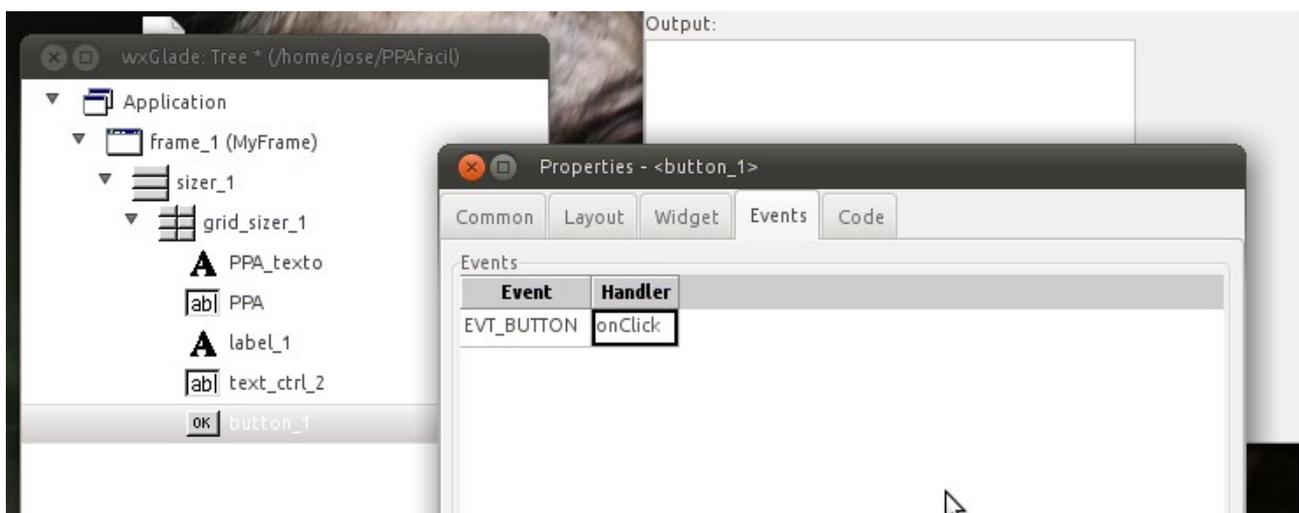


Si necesitas cambiar el nombre de la clase, lo puedes hacer desde la vista de propiedades, aquí por ejemplo selecciono el control de texto y lo renombro a “PPA” para que luego sea mas facil de usar desde el codigo.



Vale, pues con esto y la vista de Preview y jugando un poquito, no debe de ser dificil para ti ir creando una ventana con los elementos deseados. Nos queda enlazar una accion de manera que por ejemplo al pulsar un boton se ejecute “algo” que programaremos luego. Eso lo podemos hacer creando un evento.

Ejemplo: quiero que cuando pulse el boton Añadir, me lea el contenido de la caja de texto llamada “PPA” para eso, selecciono el boton, me voy a sus propiedades y selecciono la pestaña “Eventos”:



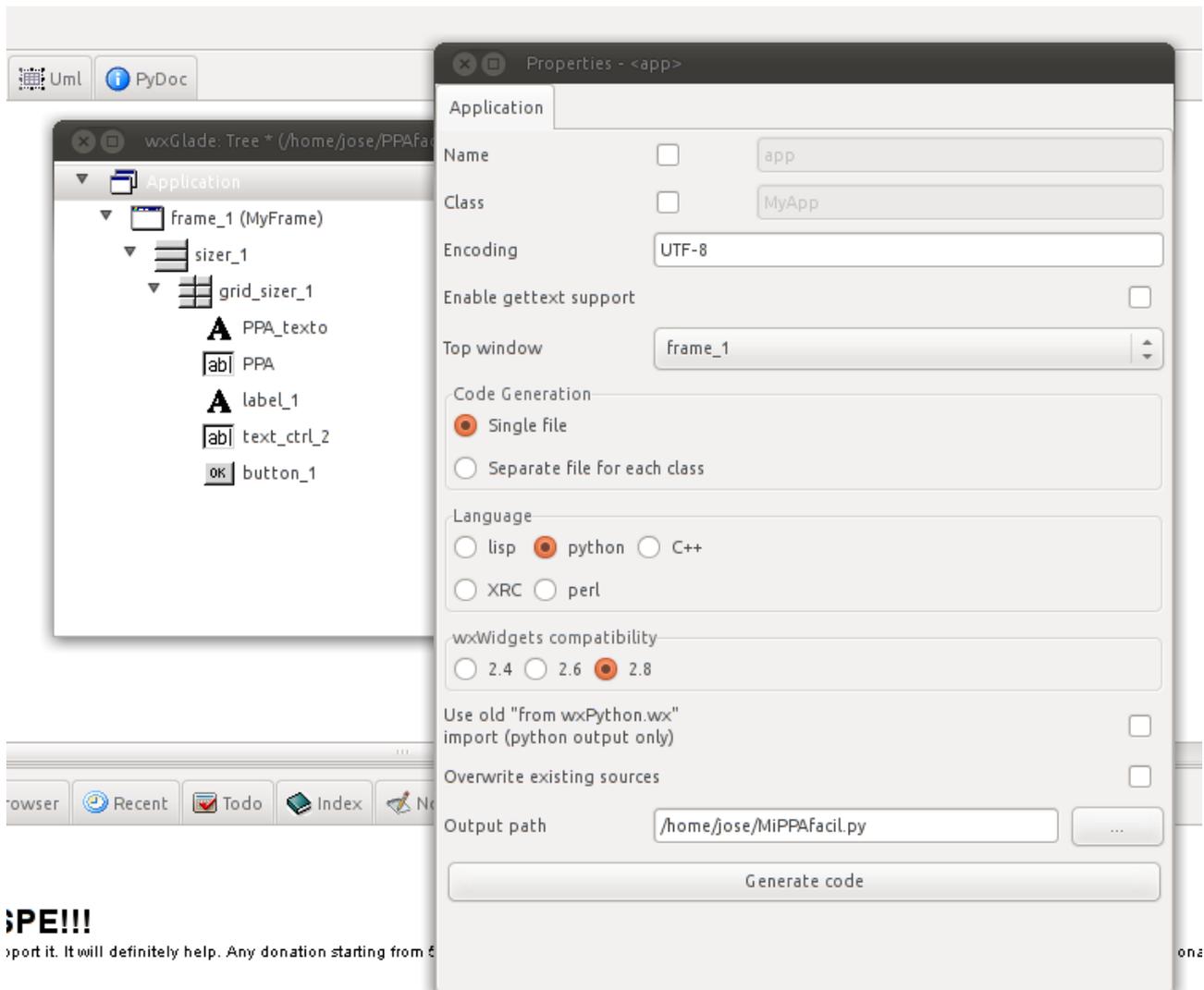
Hay un evento llamado “EVT\_BUTTON” que es precisamente el que se da cuando pulsas el boton, seleccionamos entonces al lado (Handler) y escribimos el nombre de la función que queremos que se ejecute (luego tendremos que escribir el codigo de la misma, claro). Pero yo de momento la llamo por ejemplo “onClick”, asi se que cuando pulse el boton, se ejecutara “algo” que habre

implementado con el nombre "onClick".

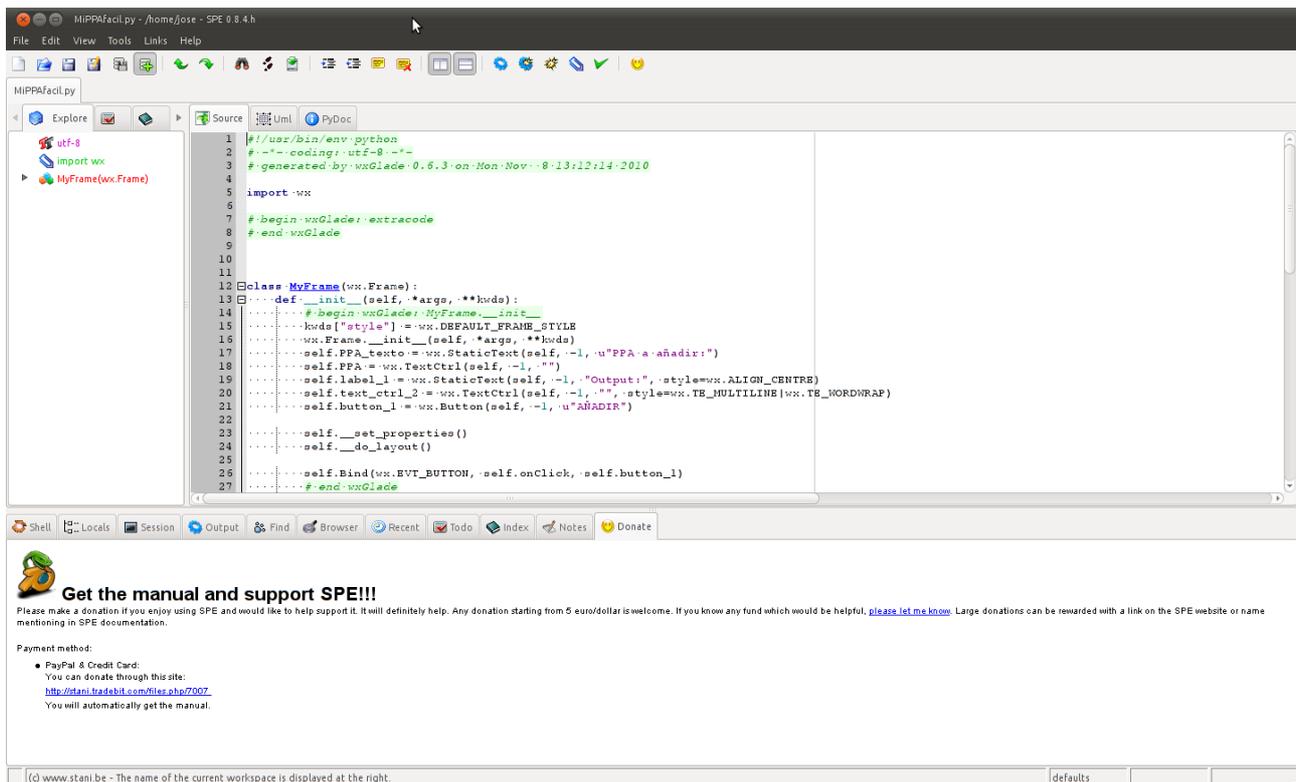
Bueno, pues termine. Tengo una ventanita muy basica, con una caja de texto donde escribire el PPA que quiero añadir, otra caja de texto donde mostrare los resultados y un boton.

Le doy a generar codigo, para ello me situo en el root del arbol, me voy a sus propiedades y configuro language="python", el path del fichero .py que quiero generar y que mapeara esta interface grafica y pulso el boton "Generate code"

:



Nos vamos ya al editor del IDE y podemos abrir el fichero generado.



Si le diéramos a ejecutar desde el IDE, haciendo click sobre la rueda dentada (o pulsando F9) directamente se mostraria la ventana grafica. Pero claro, evidentemente al pulsar el boton, no hace nada porque aun no hemos escrito el codigo del evento.

Que yo he completado con:

```
def onClick(self, event): # wxGlade: MyFrame.<event_handler>
    print "Event handler `onClick' not implemented!"

#TUTORIAL
# recuperar cadena PPA introducida
PPA_REPO=self.PPA.GetValue()
print "PPA Repo es "+PPA_REPO

# Construir el comando
COMANDO="sudo add-apt-repository "+PPA_REPO

# Ejecutar comando del sistema recuperando string
OUTPUT=""
import os
f=os.popen(COMANDO)
for i in f.readlines():
    print "myresult:",i
    OUTPUT=OUTPUT+"\r\n"+i

# Escribir output en el otro control de texto
self.text_ctrl_2.SetValue(OUTPUT)

event.Skip()
```

Deberias saber algo de python para ello, pero tampoco es complicado de seguir:

Primero recupero el contenido que escriba en la caja de texto, recuerdas aquella caja de texto que llame "PPA":

```
PPA_REPO=self.PPA.GetValue()
```

pues uso el metodo GetValue() para su nombre de clase y asi recupero el texto que cargo en una variable llamada PPA\_REPO.

Despues a ese texto le concateno: "sudo add-apt-repository" para tener el comando completo que se usa para añadir un PPA desde consola:

```
COMANDO="sudo add-apt-repository "+PPA_REPO
```

Y finalmente ejecuto ese comando, recuperando la salida y mostrandola en la caja de texto llamada "output":

```
import os  
f=os.popen(COMANDO)  
for i in f.readlines():  
    print "myresult:",i,  
    OUTPUT=OUTPUT+"\r\n"+i  
  
# Escribir output en el otro control de texto  
self.text_ctrl_2.SetValue(OUTPUT)
```

Si acaso tienes algun problema, aquí dejo el codigo completo (eso si, ten cuidado con respetar el orden de tabuladores):

```
#!/usr/bin/env python  
# -*- coding: utf-8 -*-  
# generated by wxGlade 0.6.3 on Mon Nov 8 13:12:14 2010  
  
import wx  
  
# begin wxGlade: extracode  
# end wxGlade  
  
# TUTORIAL  
import os, sys  
  
class MyFrame(wx.Frame):  
    def __init__(self, *args, **kwargs):  
        # begin wxGlade: MyFrame.__init__  
        kwargs["style"] = wx.DEFAULT_FRAME_STYLE  
        wx.Frame.__init__(self, *args, **kwargs)  
        self.PPA_texto = wx.StaticText(self, -1, u"PPA a añadir:")  
        self.PPA = wx.TextCtrl(self, -1, "")  
        self.label_1 = wx.StaticText(self, -1, "Output:", style=wx.ALIGN_CENTRE)
```

```
self.text_ctrl_2 = wx.TextCtrl(self, -1, "", style=wx.TE_MULTILINE|wx.TE_WORDWRAP)
self.button_1 = wx.Button(self, -1, u"AÑADIR")
```

```
self.__set_properties()
self.__do_layout()
```

```
self.Bind(wx.EVT_BUTTON, self.onClick, self.button_1)
# end wxGlade
```

```
def __set_properties(self):
    # begin wxGlade: MyFrame.__set_properties
    self.SetTitle("Mi add-PPA facil ")
    self.PPA_texto.SetMinSize((74, 15))
    self.PPA_texto.SetFont(wx.Font(9, wx.DEFAULT, wx.NORMAL, wx.BOLD, 0, ""))
    self.PPA.SetMinSize((280, 24))
    self.text_ctrl_2.SetMinSize((280, 160))
    # end wxGlade
```

```
def __do_layout(self):
    # begin wxGlade: MyFrame.__do_layout
    sizer_1 = wx.BoxSizer(wx.VERTICAL)
    grid_sizer_1 = wx.GridSizer(5, 1, 2, 2)
    grid_sizer_1.Add(self.PPA_texto, 0, 0, 0)
    grid_sizer_1.Add(self.PPA, 0, 0, 0)
    grid_sizer_1.Add(self.label_1, 0, 0, 0)
    grid_sizer_1.Add(self.text_ctrl_2, 0, 0, 0)
    grid_sizer_1.Add(self.button_1, 0, 0, 0)
    sizer_1.Add(grid_sizer_1, 1, wx.EXPAND, 0)
    self.SetSizer(sizer_1)
    sizer_1.Fit(self)
    self.Layout()
    # end wxGlade
```

```
def onClick(self, event): # wxGlade: MyFrame.<event_handler>
    print "Event handler `onClick' not implemented!"
```

```
#TUTORIAL
```

```
# recuperar cadena PPA introducida
PPA_REPO=self.PPA.GetValue()
print "PPA Repo es "+PPA_REPO
```

```
# Construir el comando
COMANDO="sudo add-apt-repository "+PPA_REPO
```

```
# Ejecutar comando del sistema recuperando string
OUTPUT=""
import os
f=os.popen(COMANDO)
for i in f.readlines():
    print "myresult:",i
    OUTPUT=OUTPUT+"\r\n"+i
```

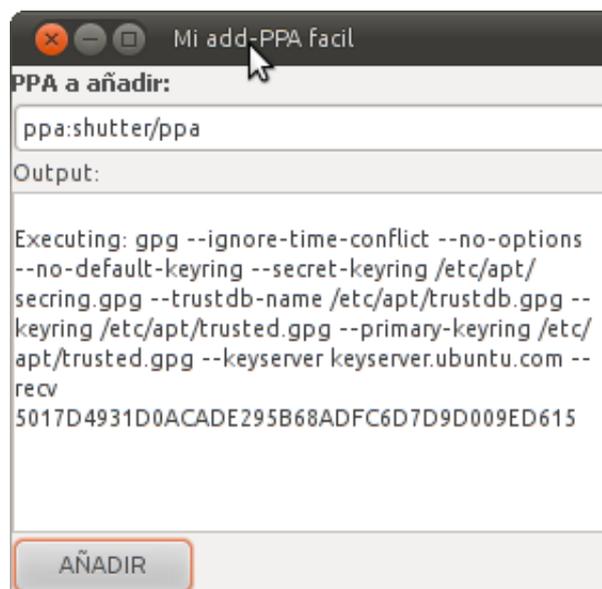
```
# Escribir output en el otro control de texto
self.text_ctrl_2.SetValue(OUTPUT)

event.Skip()
```

```
# end of class MyFrame
```

```
if __name__ == "__main__":
    app = wx.PySimpleApp(0)
    wx.InitAllImageHandlers()
    frame_1 = MyFrame(None, -1, "")
    app.SetTopWindow(frame_1)
    frame_1.Show()
    app.MainLoop()
```

Y listo, ya tenemos nuestra aplicación que tenemos que ejecutar como usuario “root” para que funcione lo de add-apt repository:



La aplicación es un poco “chorra” pero bueno, la idea era mostrar una cosa simple que sirviera de ejemplo para tus propios desarrollos.

Minitutorial escrito: el 17 noviembre 2010 por Piwi – <http://ubuntulife.wordpress.com>