

```

1  # -*-coding:utf-8 -*-
2  #qpy:kivy
3  #import sys
4  #sys.path.append('Datos')
5  #Módulo agregar usuario
6  #Desarrollado por Diego Alberto Parra Garzón
7  #Bogotá D.C., Colombia
8  # Esto es software libre licencia GPL3
9  #
10 # Para revisar si pudo agregar o no el usuario llame a la función
11 # Como_va_la_cosa()
12 # La cuál devuelve ["nada", "nada"] Si no se han llenado los text input
13 # y si los text input ya se llenaron devuelve ["valor1", "valor2"]
14 #
15
16 from kivy.app import App
17 from kivy.uix.label import Label
18 from kivy.uix.button import Button
19 from kivy.uix.relativelayout import RelativeLayout
20 from kivy.uix.textinput import TextInput
21 from kivy.clock import Clock
22
23 class ScreenAgregarUser(App):
24     #class ScreenAgregarUser(RelativeLayout):
25
26     def build(self):
27         global rl1 #Layout principal del Screen
28         global rl2 #Layout para la interaccion con el usuario
29         global txtIn1 #Text input user name
30         global txtIn2 #text input identificación user
31         global btn # Boton para agregar los usuarios
32         global lbl1 #Texto "Ingrese usuario"
33         global lbl2 #texto "Ingrese Cédula "
34         global lbl3 #texto "Advertencias que da la app "
35
36         self.comoVa = ["nada", "nada"]
37         self.Como_va_la_cosa()
38
39         rl1 = RelativeLayout()
40         rl2 = RelativeLayout()
41
42         #Inicializo los wadgets que necesita la app
43         #Inicializo los text input
44         txtIn1 = TextInput(size_hint=(0.36, 0.2), pos_hint={'center_x': .8,
45 'center_y': .95}, text="", font_size="20dp", multiline=False)
46         txtIn2 = TextInput(size_hint=(0.36, 0.2), pos_hint={'center_x': .8,
47 'center_y': .55}, text="", font_size="20dp", multiline=False,
48 input_type="number")
49
50         # Agrego los label de los text input
51         lbl1 = Label(font_size="21", pos_hint={'center_x': .8, 'center_y': .8},
52 text="Nombre completo usuario:")
53         lbl2 = Label(font_size="21", pos_hint={'center_x': .8, 'center_y': .4},
54 text="Clave dada por la empresa:")
55         lbl3 = Label(font_size="24", pos_hint={'center_x': .25,
56 'center_y': .05}, text="")
57
58         # Inicializo el boton
59         btn = Button(size_hint=(0.16, 0.2), pos_hint={'center_x': .8,
60 'center_y': .05}, background_normal="Imagenes/icons8-synchronize-64.png")
61
62         #agrego el wadget de interfaz de usuario al widget principal
63         rl1.add_widget(rl2)

```

```

58         #agrego el widget de interfaz autonoma al widget principal
59
60
61         #LLamo a la función interaccionAUTONOMA() para agregar los widget
        autonomos al widget principal
62
63
64         rl2.add_widget(lbl3)
65         self.interaccionUSER()
66         return rl1
67
68     def interaccionUSER(self):
69         rl2.add_widget(txtIn1)
70         rl2.add_widget(txtIn2)
71         rl2.add_widget(lbl1)
72         rl2.add_widget(lbl2)
73         rl2.add_widget(btn)
74         btn.bind(on_press=self.AgregarUsuario)
75
76
77     def AgregarUsuario(self, *args):
78         print "Me llamo"
79         valor1 = txtIn1.text
80         valor2 = txtIn2.text
81         if (valor1 == "") or (valor2 == ""):
82             mensaje= "Las casillas no puede estar vacias."
83             print mensaje
84             lbl3.text= mensaje
85         if (valor1 != "") and (valor2 != ""):
86             btn.unbind(on_press=self.AgregarUsuario)
87             print "Listo el nombre "
88             self.comoVa= [str(valor1), str(valor2)]
89             self.Como_va_la_cosa()
90
91             rl1.clear_widgets()
92
93
94     def Como_va_la_cosa(self):
95         comoVa1 = self.comoVa[0]
96         comoVa2 = self.comoVa[1]
97         print self.comoVa
98         return self.comoVa
99
100
101 if __name__ == "__main__":
102     ScreenAgregarUser().run()

```