

```

1  # -*-coding:utf-8 -*-
2  #::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
3  # Módulo principal de toda la aplicación de login
4  # Esto es software libre, licencia GPL3
5  # Diego Alberto Parra Garzón
6  # Bogotá D.C., Colombia
7  #::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
8  #qpy:kivy
9  #import sys
10 #sys.path.append('Datos')
11 #=====
12 # INICIALIZO LOS MÓDULOS QUE VOY A UTILIZAR
13 #=====
14
15 from kivy.uix.boxlayout import BoxLayout
16 from kivy.uix.gridlayout import GridLayout
17 from kivy.uix.relativelayout import RelativeLayout
18 from kivy.uix.textinput import TextInput
19 from kivy.app import App
20 from kivy.uix.relativelayout import RelativeLayout
21 from kivy.clock import Clock
22 from Screen1 import ScreenLogin1
23 from comUSER import CompararUsuario
24 from kivy.clock import Clock
25 from FileAgregarUser import ScreenAgregarUser
26 from kivy.uix.label import Label
27 from FileDeleteUser import ScreenDeleteUser
28 from kivy.uix.button import Button
29 #=====
30 # CREO LA CLASE LOGIN APP, QUE ES LA CLASE PRINCIPAL DE LA APLICACIÓN
31 class LoginApp(App):
32     # Definiendo las variables globales,
33     global BarAc
34     global AcView
35     global AcPre
36     global AcGroup
37     global AcBtn1
38     global AcBtn2
39     global AcBtn3
40     global screenLogin1
41     global rl
42     global rl1
43     global rl2
44     global rl3
45     global rl4
46     global rlPaso
47     global compara
48     global addUSER
49     global deleteUSER
50     global lblAlerta
51     global textIN
52     global textINpass
53     global btnCambiarpass
54     global ActionViewTitle
55     global SubBtn
56
57     # Defino los layouts que voy a utilizar
58     rl = RelativeLayout()
59     rl1 = RelativeLayout(size_hint_y=0.8, size_hint_x=1, pos=(-80,20))
60     rl2 = RelativeLayout(size_hint_y=0.7, size_hint_x=1, pos=(0,20))
61     rl3 = RelativeLayout(size_hint_y=1, size_hint_x=1, pos=(0,20))
62     rl4 = RelativeLayout()
63     rlPaso = RelativeLayout(size_hint_y=0.7, size_hint_x=0.7, pos=(150,250))
64

```

```

65
66
67     # Instancio las librerias desarrolladas para las capas de flujo de la app
68     screenLogin1 = ScreenLogin1()
69     compara = CompararUsuario()
70     addUSER = ScreenAgregarUser()
71     deleteUSER = ScreenDeleteUser()
72
73     lblAlerta = Label( pos_hint={'center_x': .7, 'center_y': .25}, text="",
font_size="20sp", markup=True)
74
75     # defino los textInput
76     textIN = TextInput(size_hint_y=0.2, size_hint_x=0.7, pos_hint=
{'center_x': .45, 'center_y': .25}, font_size="20dp", multiline=False)
77     textINpass = TextInput(size_hint_y=0.2, size_hint_x=0.5, pos_hint=
{'center_x': .45, 'center_y': .65}, font_size="20dp", multiline=False)
78
79     btnCambiarpass = Button(background_normal="Imagenes/icons8-
synchronize-64.png", pos_hint={'center_x': .85, 'center_y': .65},
size_hint_y=0.2, size_hint_x=0.2, font_size=20 )
80
81     #Defino todo lo del actionView
82     AcView = BoxLayout( pos_hint={'center_x': .5, 'center_y': .92},
orientation="horizontal")
83     AcPre = Button(background_normal="Imagenes/icons8-security-pass-64.png",
size_hint=(.13 , .13), pos_hint={'center_x': 0.9, 'center_y': .5})
84     ActionViewTitle = Label(text= "      APP DE LOGIN LooginTooth  ",
font_size="20sp")
85     AcGroup = Button(background_normal="Imagenes/icons8-settings-64.png",
size_hint=(.13 , .13), pos_hint={'center_x': 0.6, 'center_y': .5})
86
87     #defino los botones debajo del AcGroup
88     SubBtn = BoxLayout(size_hint = (0.1, 0.4), pos_hint={'center_x': .9,
'center_y': .65}, orientation = "vertical")
89     AcBtn1 = Button(background_normal="Imagenes/icons8-create-64.png",
size_hint_y= 1.6,)
90     AcBtn2 = Button(background_normal="Imagenes/icons8-trash-64.png",
size_hint_y= 1.6)
91     AcBtn3 = Button(background_normal="Imagenes/icons8-key-64.png", size_hint_y=
1.6)
92
93
94
95     def build(self):
96         # Variable para escalar permisos en la app
97         self.UsuarioROOT = 0
98
99         AcView.add_widget(AcPre)
100         AcView.add_widget(ActionViewTitle)
101         AcView.add_widget(AcGroup)
102         rl.add_widget(AcView)
103
104         rl.add_widget(lblAlerta)
105
106         SubBtn.add_widget(AcBtn1)
107         SubBtn.add_widget(AcBtn2)
108         SubBtn.add_widget(AcBtn3)
109
110         #Añadiendo los layouts de las librerias al layout principal
111         rl.add_widget(rl1)
112         rl.add_widget(rl2)
113         rl.add_widget(rl3)
114         rl.add_widget(rl4)
115         self.cuenta = 0

```

```

116         AcGroup.bind(on_press=self.identifiquese)
117
118
119
120         self.AgregarScreenComparar()
121         return rl
122
123
124
125     def QuitarScreenComparar(self):
126         try:
127             Clock.unschedule(self.EsperarIdentificacion)
128         except:
129             pass
130
131         try:
132             Clock.unschedule(self.EsperarComparacion)
133         except:
134             pass
135
136
137
138         #Función identifiquese que es llamada cuando el usuario quiere acceder a las
configuraciones
139         #de la app
140         def identifiquese(self, *args):
141             global soyROOT
142             self.QuitarScreenComparar()
143             print "Hola me llamaste para identificar el nuevo ingreso"
144             try:
145                 AcGroup.unbind(on_press=self.identifiquese)
146             except:
147                 pass
148
149             ActionViewTitle.text= "Configuraciones"
150             AcPre.background_normal="Imágenes/icons8-back-arrow-64.png"
151             try:
152                 AcPre.unbind(on_press=self.acPre1)
153             except:
154                 pass
155             try:
156                 AcPre.bind(on_press=self.acPre1)
157             except:
158                 pass
159
160             # mientras que no sea el usuario root se hará lo siguiente
161             if self.UsuarioROOT == 0 :
162                 rl1.pos=(-1000, 0)
163                 #añado un layout de paso
164                 rl.add_widget(rlPaso)
165                 rlPaso.add_widget(textIN)
166                 textIN.text=""
167                 Mensaje = "Ingrese su clave administrador"
168                 lblAlerta.pos = (0, 0)
169                 lblAlerta.pos_hint={'center_x': .5, 'center_y': .5}
170                 lblAlerta.text = Mensaje
171                 var = open("Datos/pass.text", "r")
172                 soyROOT = var.read()
173                 var.close()
174                 soyroot = soyROOT.split("\n")
175                 print soyroot[0]
176                 soyROOT = soyroot[0]
177                 Clock.schedule_interval(self.quienERES, 0.2)
178

```

```

179     # Si el usuario root se identifica la aplicación concede permisos
180     if self.UsuarioROOT == 1:
181         rl.add_widget(SubBtn)
182         AcBtn1.bind(on_press=self.acAcBtn1)
183         AcBtn2.bind(on_press=self.acAcBtn2)
184         AcBtn3.bind(on_press=self.acAcBtn3)
185         self.UsuarioROOT = 2
186
187
188     if self.UsuarioROOT == 2:
189         try:
190             rl.remove_widget(SubBtn)
191         except:
192             pass
193         try:
194             rl.add_widget(SubBtn)
195             AcBtn1.bind(on_press=self.acAcBtn1)
196             AcBtn2.bind(on_press=self.acAcBtn2)
197             AcBtn3.bind(on_press=self.acAcBtn3)
198         except:
199             pass
200
201
202     #función llamada para identificación del usuario root
203     def quienERES(self, dt):
204         print textIN.text
205         if (str(textIN.text) == (soyROOT)):
206             Clock.unschedule(self.quienERES)
207             self.UsuarioROOT = 1
208             self.valorsleep = 0
209             Mensaje = "bienvenido usuario root, espere un momento"
210             lblAlerta.pos_hint={'center_x': .5, 'center_y': .5}
211             lblAlerta.text = Mensaje
212             Clock.schedule_interval(self.timeSleep, 1)
213
214
215
216     #Este Screen es llamado en la primer vista de la app y permite
217     # el monitoreo bluetooth del lector rfid
218     def AgregarScreenComparar(self):
219         global SL
220         global sl
221         global comp
222         global cp1
223         # Clock.unschedule(self.ScreenComparar)
224         #añadiendo el screen1 al relative layout
225         var = open("Datos/SCREEN.text", "w")
226         var.write("True\n")
227         var.close()
228         SL = screenLogin1.build()
229         sl = SL.build()
230         rl1.add_widget(sl)
231
232         #iniciando la clase de comparación y añadiendola al layout principal
233         comp = compara.build()
234         rl1.add_widget(comp)
235         Clock.schedule_interval(self.EsperarIdentificacion, 0.45)
236
237
238     #=====
239     #Este Screen se ejecuta para añadir nuevos usuarios a la base de datos
240     def AgregarScreenAddUsser(self):
241         global addUser
242         addUser = addUSER.build()

```

```

243     rl1.size_hint_y = 0.6
244     rl1.size_hint_x=0.6
245     rl1.pos=(-10,100)
246     var = open("Datos/SCREEN.text", "w")
247     var.write("True\n")
248     var.close()
249     SL.ReiniciarLectura()
250     compara.CargarArray()
251     try:
252         Clock.unschedule(self.EsperarComparacion)
253     except:
254         pass
255
256     Clock.schedule_interval(self.EsperarIdentificacion1, 0.45)
257
258
259     #Este Screen se ejecuta para eliminar usuarios de la base de datos
260     def AgregarScreenDeleteUsser(self):
261         global deleteUser
262         deleteUser = deleteUser.build()
263         rl1.size_hint_y = 0.6
264         rl1.size_hint_x=0.6
265         rl1.pos=(-1500,100)
266         rl3.add_widget(deleteUser)
267         Clock.schedule_interval(self.EsperarIdentificacion2, 0.5)
268
269
270     #Este Screen es llamao para cambiar la clave del susuario root
271     def AgregarScreenCambiarPass(self):
272         print "Me llamaste para cambiar contraseñas"
273         rl1.pos=(-1000, 0)
274         rl4.add_widget(btnCambiarpass)
275         rl4.add_widget(textINpass)
276         textINpass.text = ""
277         Mensaje = "Ingrese su nueva contraseña."
278         lblAlerta.pos_hint={'center_x': .45, 'center_y': .5}
279         lblAlerta.text = Mensaje
280         btnCambiarpass.bind(on_press=self.cambiarpass)
281         pass
282
283         #Esta función es llamada para cambiar la contraseña, una vez el
284         #usuario root ga definido su nueva clave
285         def cambiarpass(self, *args):
286             #Se deja como condición que la contraseña debe ser mayor a 4 caracteres
287             if (len(textINpass.text) >= 4):
288                 btnCambiarpass.unbind(on_press=self.cambiarpass)
289                 var = open("Datos/pass.text", "w")
290                 var.write(textINpass.text)
291                 var.close()
292                 self.valorsleep = 0
293                 Mensaje = "Su contraseña de root ha sido actualizada."
294                 lblAlerta.pos_hint={'center_x': .45, 'center_y': .5}
295                 lblAlerta.text = Mensaje
296                 Clock.schedule_interval(self.timeSleep, 1)
297
298             if (len(textINpass.text) < 4):
299                 Mensaje = "\n Ingrese su nueva contraseña.\n La contraseña no puede
300 ser tan corta."
301                 lblAlerta.pos_hint={'center_x': .45, 'center_y': .5}
302                 lblAlerta.text = Mensaje
303
304         #-----
305         #-----Controla EL SCREEN DE QUITAR USUARIO

```

```

def EsperarIdentificacion2(self, dt):
    print ">>>>>>>>>>>> otro puto problema <<<<<<<<<<<<<<<<<"
+ str(deleteUSER.Asi_Va())
    if (str(deleteUSER.Asi_Va()) != "0"):
        Clock.unschedule(self.EspararIdentificacion2)
        compara.CompararArray(str(deleteUSER.Asi_Va()))
        Clock.schedule_interval(self.EspararComparacion2, 0.3)

def EspararComparacion2(self, dt):
    estado = compara.estadoCom()
    print "este es el estado del coroto " + str(estado)
    if estado != 2:
        Clock.unschedule(self.EspararComparacion2)
        self.valorsleep = 0
        Clock.schedule_interval(self.timeSleep, 1)

    pass

if estado == 2:
    print "Se encontro coincidencia deteniendo el hilo"
    Clock.unschedule(self.EspararComparacion2)
    compara.quitarUsuario()
    self.valorsleep = 0
    Clock.schedule_interval(self.timeSleep, 1)

#-----
#-----

#-----
# ----- controla EL SCREEN DE AGREGAR USUARIO
#-----

def EsperarIdentificacion1(self, dt):
    print str(SL.bufferLectura())
    if (str(SL.bufferLectura()) != "Nada"):
        Clock.unschedule(self.EspararIdentificacion1)
        compara.CompararArray(str(SL.bufferLectura()))
        Clock.schedule_interval(self.EspararComparacion1, 0.1)

def EspararComparacion1(self, dt):
    estado = compara.estadoCom()
    print "este es el estado del coroto " + str(estado)
    if estado == 1:
        pass

if estado == 2:
    print "Se encontro coincidencia deteniendo el hilo"
    Clock.unschedule(self.EspararComparacion1)
    Mensaje = " El usuario ya existe \n en el registro de la B.D."
    lblAlerta.pos_hint={'center_x': .75, 'center_y': .5}
    lblAlerta.text= Mensaje
    self.valorsleep = 0
    Clock.schedule_interval(self.timeSleep, 1)

if estado == 3:
    print "No Se encontro coincidencia deteniendo el hilo"
    Clock.unschedule(self.EspararComparacion1)
    rl2.add_widget(addUser)
    rl2.add_widget(rlPaso)
    Clock.schedule_interval(self.PreguntarFormulario, 0.4)

def PreguntarFormulario(self, dt):
```

```

368     valores = []
369     valores = addUSER.Como_va_la_cosa()
370     if(valores[1] != "nada"):
371         Clock.unschedule(self.PreguntarFormulario)
372         compara.agregarUsuario(str(SL.bufferLectura()), valores[0], valores
[1])
373         Mensaje = "      Usuario agregado al \n      registro."
374         lblAlerta.pos_hint={ 'center_x': .75, 'center_y': .5}
375         lblAlerta.text= Mensaje
376         self.valorsleep = 0
377         Clock.schedule_interval(self.timeSleep, 1)
378     #------
379     #------
380
381     #------
382     #------ controla EL SCREEN DE LOGIN
383
384     def EsperarIdentificacion(self, dt):
385         print str(SL.bufferLectura())
386         if (str(SL.bufferLectura()) != "Nada"):
387             Clock.unschedule(self.EsperarIdentificacion)
388             compara.CompararArray(str(SL.bufferLectura()))
389             Clock.schedule_interval(self.EsperarComparacion, 0.1)
390
391     def EsperarComparacion(self, dt):
392         estado = compara.estadoCom()
393         print "este es el estado del coroto " + str(estado)
394         if estado == 1:
395
396             pass
397
398         if estado == 2:
399             print "Se encontro coincidencia deteniendo el hilo"
400             Clock.unschedule(self.EsperarComparacion)
401             self.ClaveEncontrada()
402
403         if estado == 3:
404             print "No Se encontro coincidencia deteniendo el hilo"
405             Clock.unschedule(self.EsperarComparacion)
406             SL.ReiniciarLectura()
407             Clock.schedule_interval(self.EsperarIdentificacion, 0.1)
408
409     #------
410     #------
411
412     #------
413     #------Defino la función clave encontrada
414
415     def ClaveEncontrada(self):
416         self.timexit = 0
417         import time
418         infoTIME = str(time.strftime("%c"))
419         msj1 = str(SL.bufferLectura()) + " " + infoTIME + "\n"
420         print infoTIME
421         try:
422             print "=====
423             print "=====
424             print "=====
425             print "=====
426             print "=====
427             print
428             svar = "Datos/LoginApp.text"
429             sd = open(svar, "w")

```

```

430         sd.write("True")
431         sd.close()
432         svar2 = "Datos/infoLoginApp.text"
433         sd1 = open(svar2, "a")
434         sd1.write(msj1)
435         sd.close()
436         print msj1
437         print "=====
438         print "=====
439         print "=====
440         print "=====
441         print "=====
442     except:
443         pass
444     try:
445         from jnius import autoclass
446         Env = autoclass('android.os.Environment')
447         env = Env
448         print "=====
449         print "=====
450         print "=====
451         print "=====
452         print "=====
453         print "=====
454         svar = "../../../../../storage/sdcard0/"+env.DIRECTORY_DOWNLOADS + "/"
LoginApp.text"
455         sd = open(svar, "w")
456         sd.write("True")
457         sd.close()
458         svar2 = "../../../../../storage/sdcard0/"+env.DIRECTORY_DOWNLOADS +
"/infoLoginApp.text"
459         sd1 = open(svar2, "a")
460         sd1.write(msj1)
461         sd.close()
462         print "=====
463         print "=====
464         print "=====
465         print "=====
466         print "=====
467         Clock.schedule_interval(self.timeExit, 1)
468         rll.pos=(-1000, 0)
469         textINpass.text = ""
470         Mensaje = "Usuario aceptado, cerrando la App.\n Espere por favor ..."
471         lblAlerta.pos_hint={'center_x': .45, 'center_y': .5}
472         lblAlerta.text = Mensaje
473
474     except:
475         rll.pos=(-1000, 0)
476         textINpass.text = ""
477         Mensaje = " Usuario aceptado, cerrando la App. \n Espere por
favor ..."
478         lblAlerta.pos_hint={'center_x': .45, 'center_y': .5}
479         lblAlerta.text = Mensaje
480         Clock.schedule_interval(self.timeExit, 1)
481         # LoginApp().exit()
482
483         # print env.DIRECTORY_DOWNLOADS
484         pass
485
486     def timeExit(self, dt):
487         if self.timexit == 4:
488             LoginApp().get_running_app().stop()
489         else:
490             self.timexit = self.timexit + 1

```



```

491
492 #-----
493
494
495
496 #-----
497 #----- FUNCIÓN DORMIR, quita todos los procesos y vuelve al
login--
498     def timeSleep(self, dt):
499         if self.valorsleep == 1:
500             Clock.unschedule(self.timeSleep)
501             try:
502                 AcGroup.bind(on_press=self.identifiquese)
503             except:
504                 pass
505
506             try:
507                 Clock.unschedule(self.quienERES)
508             except:
509                 pass
510             try:
511                 rlPaso.clear_widgets()
512             except:
513                 pass
514             try:
515                 rl.remove_widget(rlPaso)
516             except:
517                 pass
518             try:
519                 Clock.unschedule(self.EsperarIdentificacion)
520             except:
521                 pass
522             try:
523                 Clock.unschedule(self.EsperarComparacion)
524             except:
525                 pass
526             try:
527                 Clock.unschedule(self.EsperarIdentificacion1)
528             except:
529                 pass
530             try:
531                 Clock.unschedule(self.EsperarComparacion1)
532             except:
533                 pass
534
535             try:
536                 Clock.unschedule(self.EsperarIdentificacion2)
537             except:
538                 pass
539             try:
540                 Clock.unschedule(self.EsperarComparacion2)
541             except:
542                 pass
543             try:
544                 Clock.unschedule(self.PreguntarFormulario)
545             except:
546                 pass
547
548             try:
549                 rl3.clear_widgets()
550             except:
551                 pass
552             try:
553                 rl2.clear_widgets()

```

[illegible]

```

616
617         rl.remove_widget(SubBtn)
618         ActionViewTitle.text= "Agregar nuevo usuario"
619         AcPre.background_normal="Imagenes/icons8-back-arrow-64.png"
620
621         #Añadir una acción a la imagen de ir al widget actionprevious
622         try:
623             AcPre.unbind(on_press=self.acPre1)
624         except:
625             pass
626         try:
627             AcPre.bind(on_press=self.acPre1)
628         except:
629             pass
630         AcBtn1.unbind(on_press=self.acAcBtn1)
631         AcBtn2.unbind(on_press=self.acAcBtn2)
632         AcBtn3.unbind(on_press=self.acAcBtn3)
633
634         pass
635
636         #-----Botón para quitar usuario
637         -----
638         def acAcBtn2(self, *args):
639             print "iniciando el screen de quitar usuario"
640             self.QuitarScreenComparar()
641             self.AgregarScreenDeleteUser()
642             rl.remove_widget(SubBtn)
643             ActionViewTitle.text= "Eliminar usuario"
644             AcPre.background_normal="Imagenes/icons8-back-arrow-64.png"
645             try:
646                 AcPre.unbind(on_press=self.acPre1)
647             except:
648                 pass
649             try:
650                 AcPre.bind(on_press=self.acPre1)
651             except:
652                 pass
653             AcBtn1.unbind(on_press=self.acAcBtn1)
654             AcBtn2.unbind(on_press=self.acAcBtn2)
655             AcBtn3.unbind(on_press=self.acAcBtn3)
656
657             pass
658
659         #----- Botón para cambiar contraseña
660         -----
661         def acAcBtn3(self, *args):
662             print "iniciando el screen de actualizar usuario"
663             self.QuitarScreenComparar()
664             self.AgregarScreenCambiarPass()
665             rl.remove_widget(SubBtn)
666
667             ActionViewTitle.text= "Cambia tu contraseña de administrador"
668             AcPre.background_normal="Imagenes/icons8-back-arrow-64.png"
669             #Añadir una acción a la imagen de ir al widget actionprevious
670             try:
671                 AcPre.unbind(on_press=self.acPre1)
672             except:
673                 pass
674             try:
675                 AcPre.bind(on_press=self.acPre1)
676             except:
677                 pass
678             AcBtn1.unbind(on_press=self.acAcBtn1)

```

```
678         AcBtn2.unbind(on_press=self.acAcBtn2)
679         AcBtn3.unbind(on_press=self.acAcBtn3)
680
681         pass
682
683
684
685 if __name__ == "__main__":
686     LoginApp().run()
```