

Mi diario Python

Diario sobre el aprendizaie

HOME

EJERCICIOS

MODULOS

TUTORIAL PYTHON

JUEGOS

TIPS

Aplicacion grafica con pyqt4 y sqlite3 en python

By Diego Caraballo | 10:41

5 Comentarios

E-mail Newsletter

Registra tu correo para recibir las noticias de ultima hora

OK



Madrid - Stuttgart

desde 49.00 €

IBERIA
EXPRESS

Reserva Hoy!

En
la



Conexión pyqt4 y sqlite3

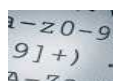
entrada anterior estuvimos viendo como crear una [aplicación en Qt Designer y Python](#). También deje unos [ejercicios para practicar con las aplicaciones gráficas](#). Hoy la idea es crear una sencilla aplicación gráfica (también utilizando pyqt4 y python) y conectar dicha aplicación a una base de datos sqlite3 para que nuestros datos sean guardados y recuperados.

Funcionalidad de la aplicación:

En la aplicación se podrán guardar los datos de los clientes (Nombre, Apellido y Localidad). El cliente será guardado en una base de datos al presionar el botón **Guardar**.
Para ver todos los clientes almacenados en nuestra base de datos presionamos el botón **Cargar**.



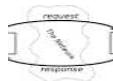
Combinar 2 combobox en PyQt - Python



Introduccion a las expresiones regulares en python



Dreampie python interactivo.



Simple programa Cliente/Servidor (socket) en python



Crear un ejecutable .exe en python con cx_Freeze

Destacado

Tutorial python desde cero --> 2.7

Ejercicios de programación python

Seguir a @PythonDiario

Contribuyentes

Jesus Becerril

Diego Caraballo

Componentes de la aplicación:

Abrimos QtDesigner y creamos un nuevo Main Window.
Tratar de dejarlo lo más parecido a la siguiente imagen:



Componentes de la aplicación

- 4 - Etiquetas (Clientes, Nombre, Apellido, Localidad)
- 3 - LineEdit
- 2 - Botones (Guardar y Cargar)
- 1 - List Widget

ObjectName de los componentes:

Son los nombres con los que haremos referencia en el código

LineEdit - (lineEdit, lineEdit_2, lineEdit_3)

Botones - (btn_Guardar, btn_Cargar)

List Widget - (lista)

Guardamos el archivo con el nombre: conBase.ui

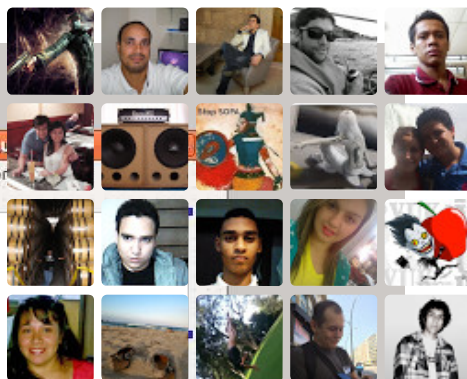
Creamos un nuevo archivo conBase.py en el [IDE](#) que nos guste más (tiene que estar en el mismo directorio que conBase.ui) y codeamos lo siguiente:

Código de la aplicación:

Google+ Seguidores

Mi diario Python

Soy seguidor/a



319 nos tienen en sus círculos.

[Ver todo](#)

Tu Contribución Permite Que Este Sitio Continúe En Funcionamiento

Donar



Buscamos Bloggeros!!!

Si te gusta nuestro Blog y deseas participar en él, te invitamos a ser Bloggero!

Para más informacion haz Clic [Aquí](#)

```

2  # -*- coding: utf-8 -*-
3
4  # Aplicacion grafica con sqlite3
5  # www.pythondiario.com
6
7  import sys
8  import sqlite3
9  from PyQt4 import QtCore, QtGui, uic
10
11 # Cargar nuestro archivo .ui
12 form_class = uic.loadUiType("conBase.ui")
13
14 class MyWindowClass(QtGui.QMainWindow, fc
15 def __init__(self, parent=None):
16     QtGui.QMainWindow.__init__(self, parent
17     self.setupUi(self)
18     self.btn_Guardar.clicked.connect(self.b
19     self.btn_Cargar.clicked.connect(self.bt
20     self.IniciarBase()
21
22
23 def IniciarBase(self):
24     self.con = sqlite3.connect("/home/diegc
25     self.cursor = self.con.cursor()
26     self.cursor.execute("""CREATE TABLE IF
27     self.con.commit()
28
29 # Evento del boton Guardar
30 def btn_Guardar_clicked(self):
31
32     self.con = sqlite3.connect("/home/diegc
33     self.cursor = self.con.cursor()
34
35 # Datos
36 self.nombre = str(self.lineEdit.text())
37 self.apellido = str(self.lineEdit_2.tex
38 self.localidad = str(self.lineEdit_3.te
39 self.datos = (self.nombre, self.apellid
40
41 # Inserta los datos en la tabla campos
42 self.cursor.execute("INSERT INTO campos
43 self.con.commit()
44
45 # Quedan los campos vacios al guardar c
46 self.lineEdit.setText("")
47 self.lineEdit_2.setText("")
48 self.lineEdit_3.setText("")
49
50 self.con.commit()
51 self.con.close()
52
53 # Evento del boton Caragar
54 def btn_Cargar_clicked(self):
55
56     self.con = sqlite3.connect("/home/diegc
57     self.cursor = self.con.cursor()
58
59 # Se cargan los datos indicados de la t
60 self.cursor.execute("SELECT NOMBRE, APE
61
62 # Al presionar el boton lo primero es b
63 self.lista.clear()
64
65 # Se agregan los elementos al QListWidg
66 for i in self.cursor:
67     self.nombre = str(i[0])
68     self.apellido = str(i[1])
69     self.localidad = str(i[2])
70
71     self.lista.addItem(self.nombre + " - "
72
73 self.con.commit()
74 self.con.close()

```



Entradas Populares



Ejercicios resueltos en python (Parte 1)

En esta entrada dejare mis soluciones a la primera parte de los ejercicios en Ejercicios Resueltos python , si no viste los ejercicios y...



Modulo random en python con ejemplos

Modulo Random En la sección Módulo s , intentare ir explicando cada uno de los módulos y las funciones que ellos contienen, vistos en e...



Eclipse y Pydev - Configuracion del IDE para python

Eclipse + Pydev Hoy vamos aprender a configurar eclipse + pydev para poder utilizarlo con python. Lo primero es instalar Eclipse

TOP



```

75
76
77 app = QtGui.QApplication(sys.argv)
78 MyWindow = MyWindowClass(None)
79 MyWindow.show()
80 app.exec_()

```

Análisis del código:

En la parte superior importamos la biblioteca **PyQt**, **sqlite3** y cargamos el archivo **conBase.ui** que generamos anteriormente con **QtDesigner**.

Creamos la clase **MyWindowClass** y le pasamos (**QtGui.QMainWindow** y **form_class**). El método **__init__** configura la interfaz GUI. En esta función tenemos que conectar los controladores de eventos:

```

self.btn_Guardar.clicked.connect(self.btn_Guardar_click
ed)
self.btn_Cargar.clicked.connect(self.btn_Cargar_clicked
)
self.IniciarBase()

```

El método **clicked.connect** es llamado cuando el botón es presionado.

Funciones de nuestra aplicación:

```

1 def IniciarBase(self):
2     self.con = sqlite3.connect("/home/diego1
3     self.cursor = self.con.cursor()
4     self.cursor.execute ("""CREATE TABLE IF
5     self.con.commit()

```

La función **IniciarBase()**, como su nombre lo indica, inicia la **base de datos** de nuestra aplicación, y es colocada en el método **__init__** para que se inicie cuando ejecutemos nuestro programa.

Si no existe, se crea la tabla **campos** con 3 elementos (**NOMBRE**, **APELLIDO**, **LOCALIDAD**).

Si no sabes como funciona una base de datos **Sqlite3** en python, te recomiendo que mires la siguiente entrada:

[Sqlite3 y Python](#)

```

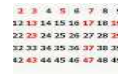
1 # Evento del boton Guardar
2 def btn_Guardar_clicked(self):
3

```



Python y SQLite3 como base de datos

Sqlite3 y Python Hoy voy a explicar lo que he ido aprendiendo sobre bases de datos en python . En este caso trabajaré con **SQLite3**, que ...



Numeros primos en python

Números primos en python He

visto que hay muchas consultas en la web relacionadas a ejercicios en python con números primos , ya sea i...



Obtener fecha y hora actual en python (datetime - time)

Fecha y hora en python ¿Como puedo saber la fecha y hora actual en python ? ¿Cuál es la función o módulo que debo usar para obtener la...



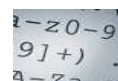
Ciclo o bucle for in en python

Ciclo o bucle for in en python Hoy vamos a ver de que trata el ciclo for in en python. En la sección Tutorial Python iré dejando todas...



Ejercicios en python (Parte 1)

Voy a dejar algunos ejercicios en python . Si hay alguna dificultad no duden en Ejercicios Python comentar. También voy a crear un post...



Introduccion a las expresiones regulares en python

Python nos permite trabajar con expresiones regulares pero para ello debemos tener antes conocimientos previos acerca de las expresi...



Como instalar python en linux

Para los que no saben, hoy vamos aprender: como instalar python en linux . Como puedo instalar python, un lenguaje orientado a objetos l...

Etiquetas

Aplicaciones Gráficas (8)

Desafíos (1)

TOP

```

4     self.con = sqlite3.connect("/home/diegc
5     self.cursor = self.con.cursor()
6
7     # Datos
8     self.nombre = str(self.lineEdit.text())
9     self.apellido = str(self.lineEdit_2.tex
10    self.localidad = str(self.lineEdit_3.te
11    self.datos = (self.nombre, self.apellid
12
13    # Inserta los datos en la tabla campos
14    self.cursor.execute("INSERT INTO campos
15    self.con.commit()
16
17    # Quedan los campos vacios al guardar c
18    self.lineEdit.setText("")
19    self.lineEdit_2.setText("")
20    self.lineEdit_3.setText("")
21
22    self.con.commit()
23    self.con.close()

```

La función `btn_Guardar_clicked(self)`, se encargará de guardar los datos del cliente en nuestra tabla `campos`. Al presionar guardar se borran los contenidos de los `lineEdit` para llenar otro cliente.

```

1     # Evento del boton Caragar
2     def btn_Cargar_clicked(self):
3
4         self.con = sqlite3.connect("/home/diegc
5         self.cursor = self.con.cursor()
6
7         # Se cargan los datos indicados de la t
8         self.cursor.execute("SELECT NOMBRE, APE
9
10        # Al presionar el boton lo primero es b
11        self.lista.clear()
12
13        # Se agregan los elementos al QListWidg
14        for i in self.cursor:
15            self.nombre = str(i[0])
16            self.apellido = str(i[1])
17            self.localidad = str(i[2])
18
19            self.lista.addItem(self.nombre + " - "
20
21        self.con.commit()
22        self.con.close()

```

La función `btn_Cargar_clicked(self)`, se encargará de mostrar en el List Widget, todos los clientes que hallamos ingresado.

Al final del código, necesitamos iniciar la interfaz gráfica:

```

1     app = QtGui.QApplication(sys.argv)
2     MyWindow = MyWindowClass(None)
3     MyWindow.show()
4     app.exec_()

```

[ejercicios \(13\)](#)

[IDE python \(2\)](#)

[juegos \(7\)](#)

[Modulos \(6\)](#)

[POO \(1\)](#)

[soluciones de ejercicios \(4\)](#)

[Tips \(10\)](#)

[Tutorial Python \(7\)](#)

Text Widget

TOP

Espero que esta entrada les sea de ayuda para sus futuras aplicaciones en python. Saludos

Like

10

Twitter

3

g+

81

Aplicaciones Gráficas base de datos

programacion orientada a objetos PyQt4

python sqlite3 qtdesigner



Author: Diego Caraballo

Hola, la idea de Python Diario es ir registrando todo lo que voy aprendiendo sobre este maravilloso lenguaje de programación. Si tienes dudas o sugerencias te pido que me lo hagas saber a través de los comentarios. Gracias



Hotel Husa de la Couronne

The hotel is located in a
From

£45.18

Book now

5 COMENTARIOS:



Anónimo 1 de diciembre de 2014, 13:56

Felicidades, lo ejecute y salio OK

Responder

Respuestas



Diego Caraballo 1 de diciembre de 2014, 14:01

Excelente :) . Saludos

TOP

[Responder](#)**Anónimo** 10 de diciembre de 2014, 6:20

Excelente! para empezar a crear proyectos...
Diego, me gustaria que compartieras un metodo de PyQt para importar imagenes, es decir asignar una foto a cada ID en la base de datos! Agradeceria la ayuda ya que la documentacion de QT esta en ingles :/ mi email es wwizka@hotmail.com

[Responder](#)

Respuestas

**Diego Caraballo** 10 de diciembre de 2014, 9:36

Hola, estuve buscando algo de información sobre como guardar imágenes en una base de datos Sqlite y encontré algunas discusiones sobre si es bueno o no guardar directamente las imágenes en la base de datos (se hace muy pesada). Algunos recomiendan usar Postgres que viene con un tipo de datos para las fotos. Otros recomiendan guardar las imágenes en una carpeta y en la base de datos guardar la ruta de dicha imagen. Quizá más adelante cree una entrada sobre este tema.
Saludos

[Responder](#)**Alexs Hm** 11 de febrero de 2015, 18:19

disculpa amigo estoy programando con python y me encontre con un problema me pide el modulo dockwidgets_rc

y la verdad no lo encuentro

[Responder](#)[TOP](#)

Introduce tu comentario...

Comentar como:

Exergy123 (Googl

Salir

Publicar

Vista previa

☐ Avisarme

Tu comentario es importante....

© 2014 [Mi diario Python](#) | Distributed By [My Blogger Themes](#) | Created By [Bloggertheme9](#)

TOP