

INFORME 1

Python y SQLite

Proyectos de Ingeniería de Software 2018
Escuela Profesional de Ingeniería de Sistemas

CARLOS CHULLO, JUAN DEYBY

Código fuente completo en el siguiente link de GITHUB: [CRUD-Phython-SQLite-Contactos](#). Nombre del archivo *Example.py*

1. Requisitos

- Python 3 o superior
- PyCharm u otro editor de texto.

2. Programación

2.1. Importaciones

```
from tkinter import *
from tkinter.ttk import *
import sqlite3
import re
```

2.2. Interfaz gráfica

Creación de la interfaz principal, se coloca: el titulo, no maximizar, icono, grilla y orientación arriba "n".

```
raiz=Tk()
raiz.title("Agenda de contactos 1.0")
raiz.resizable(0, 0)
raiz.iconbitmap("logo.ico")
miFrame=Frame()
miFrame.grid(row=0, column=0, sticky="n")
```

Variables para la asignación en la interfaz y en la base de datos

```
id_string=StringVar()
nombre_string=StringVar()
apellido_string=StringVar()
correo_string=StringVar()
telefono_string=StringVar()
estado_string=StringVar()
```

Agregando las etiquetas a la interfaz con grilla, orientación oeste y margen.

```
registro_label=Label(miFrame, text="REGISTRO")\
.grid(row=0, column=0, sticky="w", padx=5, pady=5)
id_label=Label(miFrame, text="ID :")\
.grid(row=1, column=0, sticky="w", padx=5, pady=5)
```

```

nombre_label=Label(miFrame, text="Nombre :")\
.grid(row=2, column=0, sticky="w", padx=5, pady=5)
...

```

Agregar campos para el ingreso de datos.

```

id_label=Entry(miFrame, textvariable=\
id_string, width=5)
id_label.grid(row=1, column=1, sticky="w",\
padx=5, pady=5)
id_label.config(state='readonly')
nombre_entry=Entry(miFrame, textvariable=\
nombre_string, width=23)
nombre_entry.grid(row=2, column=1, sticky="w"\
, padx=5, pady=5, columnspan=2)
...

```

2.3. Base de datos

Expresiones regulares en la entrada de datos antes procesarlos.

```

def registrar(estado_civil):
    if re.compile("^[A-z]{2,60}[\s]?+$").\
match(nombre_string.get()) is None:
        estado_string.set("Estado: Formato\
incorrecto NOMBRE")
        estado_label.config(foreground="red")
    elif re.compile("^[A-z]{2,60}[\s]?+$").\
match(apellido_string.get()) is None:
        estado_string.set("Estado: Formato \
incorrecto APELLIDO")
        estado_label.config(foreground="red")

```

Crear dos tablas con clave foránea FK base datos embebido SQLite con el nombre *usuarios.bd*. Solo una vez se ejecutara y después se debe de comentar

```

def registrar(estado_civil):
    ...
    com_bd=sqlite3.connect('usuarios.bd')
    cursor_agenda=com_bd.cursor()
    cursor_agenda.execute("CREATE TABLE "
        "estado_civil (id INTEGER PRIMARY "
        "KEY AUTOINCREMENT, "
        "nombre VARCHAR(30) UNIQUE)")
    cursor_agenda.execute("CREATE TABLE "
        "usuario (id INTEGER PRIMARY KEY "

```

```

"AUTOINCREMENT, nombre VARCHAR(30) UNIQUE,"
"apellido VARCHAR(40), estado_civil "
"INTEGER, correo VARCHAR(40), "
"telefono INT(9),"
"FOREIGN KEY (estado_civil) REFERENCES "
"estado_civil(nombre))"
com_bd.commit()
...

```

Llenado de la tabla *estado_civil*. También se debe de comentar una vez ejecutado

```

def registrar(estado_civil):
    ...
    cursor_agenda.execute("INSERT INTO "
"estado_civil (nombre) VALUES('soltero')")
    cursor_agenda.execute("INSERT INTO "
"estado_civil (nombre) VALUES('casado')")
    com_bd.commit()
    ...
    ...

```

Registrar de datos.

```

def registrar(estado_civil):
    ...
    reg=(nombre_string.get(), \
    apellido_string.get(), correo_string\
    .get(), telefono_string.get(), \
    estado_civil_value)
    com_bd=sqlite3.connect('usuarios.bd')
    cursor_agenda=com_bd.cursor()
    try:
        cursor_agenda.execute("INSERT INTO "
"usuario (nombre, apellido, correo, "
"telefono, estado_civil) VALUES(?, ?, "
" ?, ?, ?)", reg)
    except sqlite3.IntegrityError:
        estado_string.set("Estado: Ya existe"
" usuario")
        estado_label.config(foreground="red")
        return
    com_bd.commit()
    estado_string.set("Estado: Registro "
"exitoso")
    estado_label.config(foreground="green")

```

```
...  
...
```

Funciones similares al *registrar()*

```
...  
def seleccionar(e):  
...  
def eliminar():  
...  
def cargar():  
...  
def modificar():  
...  
def limpiar():  
...
```

2.4. Eventos

Botones para cada función.

```
registrar_button=Button(button_Frame, \  
text="Registrar", command=lambda : \  
registrar(estado_civil))  
registrar_button.grid(row=0, column=0, \  
padx=5, pady=5)  
  
modificar_button=Button(button_Frame, \  
text="Modificar", command=modificar)  
modificar_button.grid(row=0, column=1, \  
padx=5, pady=5)  
  
eliminar_button=Button(button_Frame, \  
text="Eliminar", command=eliminar)  
eliminar_button.grid(row=0, column=2, \  
padx=5, pady=5)
```

3. Directorio y ejecución

Name ^	Status	Date modified	Type	Size
.idea	✓	4/13/2018 7:37 PM	File folder	
venv	✓	4/2/2018 12:17 PM	File folder	
PC Example	✓	4/2/2018 12:15 PM	JetBrains PyChar...	13 KB
Example	✓	4/2/2018 12:15 PM	Python Zip Applic...	13 KB
Informe	✓	4/6/2018 10:31 AM	Microsoft Word D...	113 KB
logo	✓	4/2/2018 11:55 AM	Icon	203 KB
photo	✓	3/26/2018 4:00 PM	PNG File	9 KB
usuarios.bd	✓	4/13/2018 7:39 PM	BD File	24 KB

Figura 1: Directorio de los archivos python y SQLite

Agenta de contactos 1.0

REGISTRO

ID :

Nombre :

Apellido :

Correo :

Telefono o celular:


Estado civil :

Registrar Modificar Eliminar Cargar Limpiar

LISTA

Marielena
Mike
Mario

INFORMACION



Nombre : Marielena

Apellido : Vilca Quispe

Correo : marielena.vq@gmail.com

Telefono : 958956541

Estado civil : soltero

Estado: Destalles del contacto

Figura 2: Programa en ejecución