



Prueba 1 - Practica

11/19/2020

Objetivo:

- Consolidar los conocimientos adquiridos en clase para la programación de una aplicación en Python.

Enunciado:

- Realizar una aplicación que permita gestionar los pedidos de una empresa almacenando la información dentro de archivos o base de datos (SQLite), para lo cual deben seguir los siguientes pasos:

La aplicación deberá: manejar clientes (se guarda su nombre, dirección, teléfono y e-mail), que pueden realizar pedidos de productos, de los cuales se anota la cantidad en stock. Un cliente puede tener una o varias cuentas para el pago de los pedidos. Cada cuenta está asociada a una tarjeta de crédito, y tiene una cierta cantidad disponible de dinero, que el cliente debe aumentar periódicamente para poder realizar nuevos pedidos.

Un cliente puede empezar a realizar un pedido sólo si tiene alguna cuenta con dinero disponible. Al realizar un pedido, un cliente puede agruparlos en pedidos simples o compuestos. A su vez, un pedido compuesto contiene dos o más pedidos, que pueden ser simples o compuestos. Como es de esperar, el sistema debe garantizar que todos los pedidos simples que componen un pedido compuesto se paguen con cuentas del mismo cliente. Además, sólo es posible realizar peticiones de productos en stock.

Existe una clase (de la cual debe haber una única instancia en la aplicación) responsable del cobro, orden de distribución y confirmación de los pedidos. El cobro de los pedidos se hace una vez al día, y el proceso consiste en comprobar todos los pedidos pendientes de cobro, y cobrarlos de la cuenta de pago correspondiente. Si una cuenta no tiene suficiente dinero, el pedido se rechaza (si es parte de un pedido compuesto, se rechaza el pedido entero). Una vez que el pedido está listo para servirse, se ordena su distribución, y una vez entregado, pasa a estar confirmado.

Se aprobará como puntos adicionales a prácticas si se realiza una implementación visual utilizando cualquier librería GUI (Tkinter).

Finalmente, exportar un PDF del cuaderno de Jupyter Notebook visualizando el funcionamiento y validación del sistema.

Plazo: Se debe presentar el sistema funcionando hasta las **08:00 del 20/11/2020**, la misma que deberá ser subida al git personal y adicionalmente dentro de un cuaderno de Jupyter Notebook.



Prueba 1 - Practica

11/19/2020

Funcionamiento:

Cliente:

- Registrar Cliente

Nombre	<input type="text" value="Pedro"/>
Dirección	<input type="text" value="Carolina"/>
Teléfono	<input type="text" value="2900295"/>
E-mail	<input type="text" value="dro@gmail.com"/>
<input type="button" value="Guardar Cliente"/>	<input type="button" value="Cancelar Cliente"/>

- Consultar y agregar saldo

Saldo	<input type="text" value="200"/>				
Tarjeta Credito	<input type="text" value="23"/>				
Nombre Cliente	<input type="text" value="Pedro"/>	<input type="button" value="Buscar Cliente"/>	Dirección Cliente	<input type="text" value="Carolina"/>	
<input type="button" value="Guardar Cuenta"/>	<input type="button" value="Cancelar Cuenta"/>		Teléfono Cliente	<input type="text" value="2900295"/>	
Nombre Producto	<input type="text"/>		Email Cliente	<input type="text" value="pedro@gmail.co"/>	
Precio Producto	<input type="text"/>				

- Recarga

		Pedidos	
Nombre Cliente	<input type="text" value="Pedro"/>	<input type="button" value="Buscar Cliente Pedido"/>	
Saldo	<input type="text" value="200"/>		
Dirección Cliente	<input type="text" value="Carolina"/>		
Teléfono Cliente	<input type="text" value="2900295"/>		
Email Cliente	<input type="text" value="pedro@gmail.co"/>		
Saldo a Recargar	<input type="text" value="100"/>	<input type="button" value="Recargar"/>	



Prueba 1 - Practica

11/19/2020

Pedidos		
Nombre Cliente	<input type="text" value="Pedro"/>	<input type="button" value="Buscar Cliente Pedido"/>
Saldo	<input type="text" value="300.0"/>	
Dirección Cliente	<input type="text" value="Carolina"/>	
Teléfono Cliente	<input type="text" value="2900295"/>	
Email Cliente	<input type="text" value="pedro@gmail.c"/>	
Saldo a Recargar	<input type="text"/>	<input type="button" value="Recargar"/>

Productos:

- Registrar Productos

Nombre Producto	<input type="text" value="Fideo"/>
Precio Producto	<input type="text" value="1.20"/>
Stock Producto	<input type="text" value="100"/>
<input type="button" value="Guardar Producto"/>	<input type="button" value="Cancelar Producto"/>

Pedidos:

- Realizar pedido

Pedidos		Productos			
Nombre Cliente	<input type="text" value="Pedro"/>	<input type="button" value="Buscar Cliente Pedido"/>	Nombre Producto	<input type="text" value="Fideo"/>	<input type="button" value="Buscar Producto"/>
Saldo	<input type="text" value="300"/>		Stock	<input type="text" value="40"/>	
Dirección Cliente	<input type="text" value="Carolina"/>		Precio	<input type="text" value="1.2"/>	
Teléfono Cliente	<input type="text" value="2900295"/>				
Email Cliente	<input type="text" value="pedro@gmail.c"/>		Cantidad	<input type="text" value="10"/>	<input type="button" value="Calcular"/>
			Total	<input type="text" value="12.0"/>	<input type="button" value="Comprar"/>
Saldo a Recargar	<input type="text"/>	<input type="button" value="Recargar"/>			

- Cambio de stock y saldo, despues de la compra



11/19/2020

Interfaz Compleeta:

Nombre

Dirección

Teléfono

E-mail

Guardar Cliente

Cancelar Cliente

Saldo

Tarjeta Credito

Nombre Cliente

Buscar Cliente

Dirección Cliente

Guardar Cuenta

Cancelar Cuenta

Teléfono Cliente

Nombre Producto

Email Cliente

Precio Producto

Stock Producto

Guardar Producto

Cancelar Producto

Nombre Cliente

Pedro

Saldo

288

Dirección Cliente

Carolina

Teléfono Cliente

2900295

Email Cliente

pedro@gmail.com

Buscar Cliente Pedido

Nombre Producto

Fideo

Stock

30

Precio

1.2

Buscar Producto

Saldo a Recargar

Recargar

Cantidad

Calcular

Total

Comprar

```
In [*]: from tkinter import *
from tkinter import messagebox as MessageBox
import sqlite3

connection = sqlite3.connect('Pedidos.db')

ventana = Tk()

ventana.title("Pedidos")

ventana.geometry('1000x700')

lbl = Label(ventana, text="Nombre")

lbl.grid(column=0, row=0)

lbl1 = Label(ventana, text="Dirección")

lbl1.grid(column=0, row=1)

lbl2 = Label(ventana, text="Teléfono")

lbl2.grid(column=0, row=2)

lbl3 = Label(ventana, text="E-mail")

lbl3.grid(column=0, row=3)

lbl4 = Label(ventana, text="Saldo")

lbl4.grid(column=0, row=5)

lbl5 = Label(ventana, text="Tarjeta Credito")

lbl5.grid(column=0, row=6)

lbl6 = Label(ventana, text="Nombre Cliente")

lbl6.grid(column=0, row=7)

lbl6 = Label(ventana, text="Dirección Cliente")

lbl6.grid(column=3, row=7)

lbl7 = Label(ventana, text="Teléfono Cliente")

lbl7.grid(column=3, row=8)

lbl8 = Label(ventana, text="Email Cliente")

lbl8.grid(column=3, row=9)
```

```
lbl9 = Label(ventana, text="Nombre Producto")
lbl9.grid(column=0, row=9)

lbl10 = Label(ventana, text="Precio Producto")
lbl10.grid(column=0, row=10)

lbl11 = Label(ventana, text="Stock Producto")
lbl11.grid(column=0, row=11)

lbl12 = Label(ventana, text="-----")
lbl12.grid(column=0, row=13)

lbl13 = Label(ventana, text="-----")
lbl13.grid(column=1, row=13)

lbl14 = Label(ventana, text="Pedidos")
lbl14.grid(column=2, row=13)

lbl15 = Label(ventana, text="-----")
lbl15.grid(column=3, row=13)

lbl16 = Label(ventana, text="-----")
lbl16.grid(column=4, row=13)

lbl17 = Label(ventana, text="Nombre Cliente")
lbl17.grid(column=0, row=14)

lbl18 = Label(ventana, text="Nombre Producto")
lbl18.grid(column=3, row=14)

lbl19 = Label(ventana, text="Saldo")
lbl19.grid(column=0, row=15)

lbl20 = Label(ventana, text="Stock")
lbl20.grid(column=3, row=15)

lbl21 = Label(ventana, text="Precio")
lbl21.grid(column=3, row=16)

lbl22 = Label(ventana, text="Dirección Cliente")
```

```
lbl22.grid(column=0, row=16)

lbl23 = Label(ventana, text="Teléfono Cliente")
lbl23.grid(column=0, row=17)

lbl24 = Label(ventana, text="Email Cliente")
lbl24.grid(column=0, row=18)

lbl25 = Label(ventana, text="Cantidad")
lbl25.grid(column=3, row=19)

lbl26 = Label(ventana, text="Total")
lbl26.grid(column=3, row=20)

lbl27 = Label(ventana, text="Saldo a Recargar")
lbl27.grid(column=0, row=21)

txt = Entry(ventana,width=10,)
txt.grid(column=1, row=0)

txt1 = Entry(ventana,width=10)
txt1.grid(column=1, row=1)

txt2 = Entry(ventana,width=10)
txt2.grid(column=1, row=2)

txt3 = Entry(ventana,width=10)
txt3.grid(column=1, row=3)

txt4 = Entry(ventana,width=10)
txt4.grid(column=1, row=5)

txt5 = Entry(ventana,width=10)
txt5.grid(column=1, row=6)

txt6 = Entry(ventana,width=10)
txt6.grid(column=1, row=7)

txt7 = Entry(ventana,width=10)
txt7.grid(column=4, row=7)
```

```
txt8 = Entry(ventana,width=10)
txt8.grid(column=4, row=8)
txt9 = Entry(ventana,width=10)
txt9.grid(column=4, row=9)
txt10 = Entry(ventana,width=10)

txt11 = Entry(ventana,width=10)
txt11.grid(column=1, row=9)
txt12 = Entry(ventana,width=10)
txt12.grid(column=1, row=10)
txt13 = Entry(ventana,width=10)
txt13.grid(column=1, row=11)
txt14 = Entry(ventana,width=10)
txt14.grid(column=1, row=14)
txt15 = Entry(ventana,width=10)
txt15.grid(column=4, row=14)
txt16 = Entry(ventana,width=10)
txt16.grid(column=1, row=15)
txt17 = Entry(ventana,width=10)
txt17.grid(column=4, row=15)
txt18 = Entry(ventana,width=10)
txt18.grid(column=4, row=16)
txt19 = Entry(ventana,width=10)
txt19.grid(column=1, row=16)
txt20 = Entry(ventana,width=10)
txt20.grid(column=1, row=17)
txt21 = Entry(ventana,width=10)
txt21.grid(column=1, row=18)
```



```
txt22 = Entry(ventana,width=10)
txt22.grid(column=4, row=19)
txt23 = Entry(ventana,width=10)
txt23.grid(column=4, row=20)
txt24 = Entry(ventana,width=10)
txt25 = Entry(ventana,width=10)
txt26 = Entry(ventana,width=10)
txt26.grid(column=1, row=21)

var="null"
auxCliente=0

def sql_insertar_Cliente(con, entities):
    cursorObj = con.cursor()
    cursorObj.execute('INSERT INTO cliente(nombre, dirección, telef
    con.commit()

def guardarCliente():
    con = sqlite3.connect('Pedidos.db')
    entities = (txt.get(), txt1.get(), txt2.get(), txt3.get())
    sql_insertar_Cliente(con, entities)
    txt.delete(0, END)
    txt1.delete(0, END)
    txt2.delete(0, END)
    txt3.delete(0, END)

btn = Button(ventana, text="Guardar Cliente", command=guardarClient
btn.grid(column=0, row=4)

def clickedCancelar():
    txt.delete(0, END)
    txt1.delete(0, END)
    txt2.delete(0, END)
    txt3.delete(0, END)
```

```
btn1 = Button(ventana, text="Cancelar Cliente", command=clickedCanc
btn1.grid(column=1, row=4)

def sql_datos_Cliente():

    con = sqlite3.connect('Pedidos.db')

    cursorObj = con.cursor()
    primera='select * from cliente where nombre = "'
    nombre= ""
    nombre=txt6.get()
    consult = primera+nombre+'"'

    cursorObj.execute(consult)

    rows = cursorObj.fetchall()

    for row in rows:

        txt7.insert(0, row[2])
        txt8.insert(0, row[3])
        txt9.insert(0, row[4])
        txt10.insert(0, row[0])

btn4 = Button(ventana, text="Buscar Cliente", command=sql_datos_Cli
btn4.grid(column=2, row=7)

def sql_insertar_Cuenta(con, entities):

    cursorObj = con.cursor()

    cursorObj.execute('INSERT INTO cuenta(saldo, num_tarjeta, idcl
    con.commit()

def guardarCuenta():

    con = sqlite3.connect('Pedidos.db')

    entities = (txt4.get(), txt5.get(), txt10.get())

    sql_insertar_Cuenta(con, entities)

    txt4.delete(0, END)
    txt5.delete(0, END)
    txt6.delete(0, END)
    txt7.delete(0, END)
    txt8.delete(0, END)
    txt9.delete(0, END)
    txt10.delete(0, END)
```

```
btn2 = Button(ventana, text="Guardar Cuenta", command=guardarCuenta)
btn2.grid(column=0, row=8)

def clickedCancelarCuenta():
    txt4.delete(0, END)
    txt5.delete(0, END)
    txt6.delete(0, END)
    txt7.delete(0, END)
    txt8.delete(0, END)
    txt9.delete(0, END)
    txt10.delete(0, END)

btn3 = Button(ventana, text="Cancelar Cuenta", command=clickedCancelarCuenta)
btn3.grid(column=1, row=8)

def sql_insertar_Producto(con, entities):
    cursorObj = con.cursor()
    cursorObj.execute('INSERT INTO producto(nombre, precio, stock) VALUES(?, ?, ?)')
    con.commit()

def guardarProducto():
    con = sqlite3.connect('Pedidos.db')
    entities = (txt11.get(), txt12.get(), txt13.get())
    print(entities)
    sql_insertar_Producto(con, entities)
    txt11.delete(0, END)
    txt12.delete(0, END)
    txt13.delete(0, END)

btn4 = Button(ventana, text="Guardar Producto", command=guardarProducto)
btn4.grid(column=0, row=12)

def clickedCancelarProducto():
    txt11.delete(0, END)
    txt12.delete(0, END)
    txt13.delete(0, END)

btn5 = Button(ventana, text="Cancelar Producto", command=clickedCancelarProducto)
```

```
btn5.grid(column=1, row=12)

def sql_datos_Cliente1():

    con = sqlite3.connect('Pedidos.db')

    cursorObj = con.cursor()
    primera='select * from cliente c, cuenta cu where nombre = "'
    nombre= ""
    nombre=txt14.get()
    consult = primera+nombre+" AND c.id = cu.idcliente'

    print(consult)
    cursorObj.execute(consult)

    rows = cursorObj.fetchall()

    sald=0
    for row in rows:
        txt19.insert(0, row[2])
        txt20.insert(0, row[3])
        txt21.insert(0, row[4])
        txt16.insert(0, row[6])
        txt25.insert(0, row[5])
        sald = row[6]

    a = 1
    print(sald)
    if sald >= a:
        print("bien")

    else:

        MessageBox.showwarning("Alerta", "No tiene Saldo porfavor r
        txt19.delete(0, END)
        txt20.delete(0, END)
        txt21.delete(0, END)
        txt16.delete(0, END)
        txt14.delete(0, END)

btn6 = Button(ventana, text="Buscar Cliente Pedido", command=sql_da
btn6.grid(column=2, row=14)

def sql_datos_Cliente1():

    con = sqlite3.connect('Pedidos.db')

    cursorObj = con.cursor()
    primera='select * from producto where nombre = "'
```

```

primera= select * from productos where nombre =
nombre= ""
nombre=txt15.get()
consult = primera+nombre+""

print(consult)
cursorObj.execute(consult)

rows = cursorObj.fetchall()

for row in rows:
    txt18.insert(0, row[2])
    txt17.insert(0, row[3])
    txt24.insert(0, row[0])

btn6 = Button(ventana, text="Buscar Producto", command=sql_datos_Cl

btn6.grid(column=5, row=14)

def sumar_Cantidad():
    suma = 0
    aux1 = 0
    aux2 = 0
    aux1 = txt22.get()
    aux2 = txt18.get()
    suma = float(aux1) * float(aux2)
    txt23.insert(0, float(suma))

btn7 = Button(ventana, text="Calcular", command=sumar_Cantidad)

btn7.grid(column=5, row=19)

def sql_update_Producto():

    con = sqlite3.connect('Pedidos.db')

    cursorObj = con.cursor()
    num1 = 0
    num2 = 0
    num3 = 0
    num1 = int(txt17.get())
    num2 = int(txt22.get())
    num3 = int(num1)-int(num2)

    num11 = 0
    num21 = 0
    num31 = 0
    num11 = float(txt16.get())
    num21 = float(txt23.get())
    num31 = float(num11)-float(num21)
    tx11 = "UPDATE cuenta SET saldo = " + str(num31)
    tx21 = " where idC = " + str(txt25.get())
    tx31 = tx11 +tx21

```

```
tx1 = "UPDATE producto SET stock = " + str(num3)
tx2 = " where idProducto = " + str(txt24.get())
tx3 = tx1 +tx2
print(tx3)
cursorObj.execute(tx3)
con.commit()
txt18.delete(0, END)
txt17.delete(0, END)
txt24.delete(0, END)
txt15.delete(0, END)
txt22.delete(0, END)
txt23.delete(0, END)

cursorObj1 = con.cursor()

print(tx31)
cursorObj1.execute(tx31)
con.commit()

txt19.delete(0, END)
txt20.delete(0, END)
txt21.delete(0, END)
txt16.delete(0, END)
txt14.delete(0, END)

btn8 = Button(ventana, text="Comprar", command=sql_update_Producto)
btn8.grid(column=5, row=20)

def sql_update_Producto():

    con = sqlite3.connect('Pedidos.db')

    cursorObj1 = con.cursor()
    num11 = 0
    num21 = 0
    num31 = 0
    num11 = float(txt16.get())
    num21 = float(txt26.get())
    num31 = float(num11)+float(num21)
    tx11 = "UPDATE cuenta SET saldo = " + str(num31)
    tx21 = " where idC = " + str(txt25.get())
    tx31 = tx11 +tx21

    print(tx31)
    cursorObj1.execute(tx31)
    con.commit()

    txt16.delete(0, END)
    txt16.insert(0, num31)
    txt26.delete(0, END)
```

```
btn9 = Button(ventana, text="Recargar", command=sql_update_Producto)
btn9.grid(column=2, row=21)
ventana.mainloop()
```

```
select * from cliente c, cuenta cu where nombre = "Mateo" AND c.id
= cu.idcliente
200
bien
UPDATE cuenta SET saldo = 500.0 where idC = 1
select * from cliente c, cuenta cu where nombre = "d" AND c.id = c
u.idcliente
0
```