import xml.etree.ElementTree as ET

Código de app.py

Pregunta #4

import PySimpleGUI as sg

from graphviz import Digraph

class Producto:

    def **\_\_init\_\_**(self, nombre, precio, cantidad):

        self.nombre = nombre

        self.precio = precio

        self.cantidad = cantidad

        self.next = None

class ListaEnlazada:

    def **\_\_init\_\_**(self):

        self.head = None

    def **agregar\_producto**(self, producto):

        if not self.head:

            self.head = producto

        else:

            last\_node = self.head

            while last\_node.next:

                last\_node = last\_node.next

            last\_node.next = producto

    def **mostrar\_productos**(self):

        node = self.head

        while node:

**print**(f'Nombre: {node.nombre}, Precio: {node.precio}, Cantidad: {node.cantidad}')

            node = node.next

    def **buscar\_producto**(self, nombre):

        node = self.head

        while node:

            if node.nombre == nombre:

                return node

**print**(f'Nombre: {node.nombre}, Precio: {node.precio}, Cantidad: {node.cantidad}')

            node = node.next

        return None

    def **eliminar\_producto**(self, nombre):

        if self.head is None:

            return

        if self.head.nombre == nombre:

            self.head = self.head.next

            return

        node = self.head

        while node.next:

            if node.next.nombre == nombre:

                node.next = node.next.next

                return

            node = node.next

    def **generar\_grafo**(self):

        dot = Digraph(comment='Lista Enlazada')

        node = self.head

        i = 0

        while node:

            dot.**node**(str(i), f'Nombre: {node.nombre}\nPrecio: {node.precio}\nCantidad: {node.cantidad}')

            if node.next is not None:

                dot.**edge**(str(i), str(i+1))

            i += 1

            node = node.next

        dot.format = 'png'

        dot.**render**('lista\_enlazada.dot', view=True)

def **leer\_xml**(file):

    tree = ET.**parse**(file)

    root = tree.**getroot**()

    lista = ListaEnlazada()

    for producto in root.**findall**('producto'):

        nombre = producto.**find**('nombre').text

        precio = float(producto.**find**('precio').text)

        cantidad = int(producto.**find**('cantidad').text)

        lista.**agregar\_producto**(Producto(nombre, precio, cantidad))

    return lista

def **main**():

    lista = **leer\_xml**('productos.xml')

    layout = [[sg.Text('Menú')],

              [sg.Button('Mostrar productos')],

              [sg.Button('Buscar producto')],

              [sg.Button('Agregar producto')],

              [sg.Button('Eliminar producto')],

              [sg.Button('Generar grafo')],

              [sg.**Exit**()]]

    window = sg.Window('Menú', layout)

    while True:

        event, values = window.**read**()

        if event in (None, 'Exit'):

            break

        elif event == 'Mostrar productos':

            lista.**mostrar\_productos**()

            node = lista.head

            productos = []

            while node:

                productos.**append**(f'Nombre: {node.nombre}, Precio: {node.precio}, Cantidad: {node.cantidad}')

                node = node.next

            layout\_productos = [[sg.Text(producto)] for producto in productos]

            layout\_productos.**append**([sg.Button('Cerrar')])

            window\_productos = sg.Window('Productos', layout\_productos)

            while True:

                event\_productos, values\_productos = window\_productos.**read**()

                if event\_productos in (None, 'Cerrar'):

                    break

            window\_productos.**close**()

        elif event == 'Buscar producto':

            nombre = sg.**popup\_get\_text**('Nombre del producto')

            producto = lista.**buscar\_producto**(nombre)

            if producto:

                sg.**popup**(f'Nombre: {producto.nombre}\nPrecio: {producto.precio}\nCantidad: {producto.cantidad}')

            else:

                sg.**popup**('Producto no encontrado')

        elif event == 'Agregar producto':

            nombre = sg.**popup\_get\_text**('Nombre del producto')

            precio = float(sg.**popup\_get\_text**('Precio del producto'))

            cantidad = int(sg.**popup\_get\_text**('Cantidad del producto'))

            lista.**agregar\_producto**(Producto(nombre, precio, cantidad))

        elif event == 'Eliminar producto':

            nombre = sg.**popup\_get\_text**('Nombre del producto')

            lista.**eliminar\_producto**(nombre)

        elif event == 'Generar grafo':

            lista.**generar\_grafo**()

    window.**close**()

if \_\_name\_\_ == "\_\_main\_\_":

**main**()

Código del xml

<?xml version="1.0" encoding="UTF-8"?>

<productos>

    <producto>

        <nombre>Producto 1</nombre>

        <precio>10.99</precio>

        <cantidad>5</cantidad>

    </producto>

    <producto>

        <nombre>Producto 2</nombre>

        <precio>20.99</precio>

        <cantidad>10</cantidad>

    </producto>

    <producto>

        <nombre>Producto 3</nombre>

        <precio>15.99</precio>

        <cantidad>7</cantidad>

    </producto>

    <producto>

        <nombre>Producto 4</nombre>

        <precio>25.99</precio>

        <cantidad>9</cantidad>

    </producto>

    <producto>

        <nombre>Producto 5</nombre>

        <precio>30.99</precio>

        <cantidad>11</cantidad>

    </producto>

    <producto>

        <nombre>Producto 6</nombre>

        <precio>35.99</precio>

        <cantidad>13</cantidad>

    </producto>

    <producto>

        <nombre>Producto 7</nombre>

        <precio>40.99</precio>

        <cantidad>15</cantidad>

    </producto>

    <producto>

        <nombre>Producto 8</nombre>

        <precio>45.99</precio>

        <cantidad>17</cantidad>

    </producto>

</productos>

El programa únicamente cuenta con app.py



Y el archivo "productos.xml"



Que contiene los datos antes mencionados.

El menú de la app.py se ve de la siguiente forma:

Captura de pantalla de un celular con letras

Descripción generada automáticamente

Que nos da un total de 6 opciones.

Al seleccionar la opción Mostrar productos:

Nos muestra en Consola:

Texto

Descripción generada automáticamente

Y en una ventana.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al seleccionar la opción Buscar producto:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Recorre el xml mostrándonos nuevamente en consola los productos existentes:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Al encontrar el producto despliega la siguiente ventana

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

En cazo de no existir ningún producto con el nombre que pusimos a buscar lanza el siguiente mensaje:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al seleccionar la opción Agregar producto:

Nos permite darle nombre, precio, cantidad

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Y si recorremos de nuevo mostrar el producto confirmamos que esta agregado:

Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al seleccionar la opción Eliminar producto:

Nos muestra lo siguiente para especificar que producto vamos a eliminar

Interfaz de usuario gráfica

Descripción generada automáticamente

Aquí un ejemplo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Comprobamos si se eliminó seleccionando de nuevo mostrar producto:

Texto

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como se observa se logro eliminar.

Al seleccionar la opción Generar Grafo:

Genera un archivo .dot y un .png del grafo



Diagrama, Esquemático

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al seleccionar la opción Exit:

Este sale del programa dando como finalizado la presentación.