



Universidad
Tecnológica
de Bolívar

Práctica Estructura de Datos Python

Ejecutor Técnico

Guillermo Bejarano Reyes

INTELIGENCIA ARTIFICIAL

Nivel Explorador

www.utb.edu.co/talento-tech

Práctica Dirigida: Estructura de Datos con Python

Ejercicio Lista de Productos

Enunciado del ejercicio:

Desarrolla un programa en Python que permita gestionar una lista de productos comprados, mostrando un resumen de la compra y el total a pagar. El programa debe cumplir con los siguientes requisitos:

1.Ingreso de productos:

1. El programa debe permitir agregar múltiples productos.
2. Para cada producto, el usuario debe ingresar:
 1. Nombre del producto.
 2. Cantidad comprada.
 3. Valor unitario del producto (precio por unidad).

2.Cálculo del total por producto y total de la compra:

1. El programa debe calcular el total por cada producto, multiplicando la cantidad por el valor unitario.
2. También debe calcular y mostrar el total de la compra, sumando los totales de cada producto.

3 Opciones del menú:

- El programa debe presentar un menú con las siguientes opciones:
 - 1.Agregar un nuevo producto.
 - 2.Generar y mostrar los resultados, que incluyen el detalle de cada producto (nombre, cantidad, valor unitario, y total) y el total a pagar de toda la compra.
 - 3.Salir del programa.
- Validación de entrada:**
- El programa debe validar las opciones ingresadas en el menú y manejar errores si el usuario selecciona una opción no válida.

Ejemplo de flujo de ejecución:

- 1.El programa muestra el menú.
- 2.Si el usuario selecciona la opción "1", se le solicita la información del producto (nombre, cantidad y valor unitario) y se almacena en una lista.
- 3.Si el usuario selecciona la opción "2", el programa muestra un resumen de cada producto con su cantidad, valor unitario, y total por producto, y luego el total a pagar de toda la compra.
- 4.Si el usuario selecciona la opción "3", el programa finaliza.
- 5.Si el usuario ingresa una opción inválida, el programa muestra un mensaje de error y vuelve a mostrar el menú.

Nota: Utiliza estructuras de datos adecuadas, como listas y diccionarios, para almacenar y organizar la información de los productos.

Menú:

```
1. Agregar producto
2. Generar resultados
3. Salir
Seleccione una opción: 1
Ingrese el nombre del producto: Televisor
Ingrese la cantidad comprada: 2
Ingrese el valor unitario del producto: 200
Producto Televisor agregado con éxito.
```

Menú:

```
1. Agregar producto
2. Generar resultados
3. Salir
Seleccione una opción: 1
Ingrese el nombre del producto: Nevera
Ingrese la cantidad comprada: 1
Ingrese el valor unitario del producto: 500
Producto Nevera agregado con éxito.
```

Menú:

```
1. Agregar producto
2. Generar resultados
3. Salir
Seleccione una opción: 2
```

Resumen de la compra:

```
Producto: Televisor - Cantidad: 2 - Valor unitario: $200.0 - Total: $400.00
Producto: Nevera - Cantidad: 1 - Valor unitario: $500.0 - Total: $500.00
```

Total a pagar por la compra: \$900.00

Menú:

```
1. Agregar producto
2. Generar resultados
3. Salir
```

```
# Ciclo para el menú
while True:
    # Menú de opciones
    print("Menú:")
    print("1. Agregar producto")
    print("2. Generar resultados")
    print("3. Salir")

    # Leer la opción seleccionada
    opcion = input("Seleccione una opción: ")

    # Opción 1: Agregar un producto
    if opcion == "1":
        nombre = input("Ingrese el nombre del producto: ")
        cantidad = int(input("Ingrese la cantidad comprada: "))
        valor_unitario = float(input("Ingrese el valor unitario del producto: "))

        # Crear un diccionario con la información del producto
        producto = {
            "nombre": nombre,
            "cantidad": cantidad,
            "valor_unitario": valor_unitario
        }

        # Agregar el diccionario a la lista de productos
        lista_productos.append(producto)
        print(f"Producto {nombre} agregado con éxito.\n")
```

```
# Opción 2: Generar los resultados
elif opcion == "2":
    total_compra = 0
    print("\nResumen de la compra:")

    # Recorrer la lista de productos
    for producto in lista_productos:
        # Calcular el total por producto
        total_producto = producto["cantidad"] * producto["valor_unitario"]
        total_compra += total_producto

        # Mostrar el detalle del producto
        print(f"Producto: {producto['nombre']} - Cantidad: {producto['cantidad']} - "
              f"Valor unitario: ${producto['valor_unitario']:.2f} - Total: ${total_producto:.2f}")

    # Mostrar el total a pagar por la compra
    print(f"\nTotal a pagar por la compra: ${total_compra:.2f}\n")

# Opción 3: Salir
elif opcion == "3":
    print("Saliendo del programa...")
    break # Salir del ciclo y terminar el programa

# Si la opción no es válida
else:
    print("Opción inválida. Intente de nuevo.\n")
```


Ejercicio Comisión Vendedores

COMISIONES VENDEDORES

Dada la información de las ventas realizadas por los vendedores de una empresa, se debe ingresar el nombre del vendedor, el valor de las ventas realizadas y el tipo de vendedor (1="Puerta a puerta", 2="Telemercadeo"), a través de un diccionario y almacenarlos en una lista. Con los datos recopilados, se pide calcular e imprimir:

- Valor comisión de cada vendedor: si el tipo de vendedor es 1 (Puerta a puerta) gana el 25% de comisión sobre el valor de las ventas. Si el tipo de vendedor es 2 (Telemercadeo) gana el 15% de comisión sobre el valor de las ventas
- Valor total por pagar por comisiones

Realizar un menú con las opciones: Agregar elementos a la lista, generar los resultados y salir

COMPRAS

Desarrolla un programa en Python que permita gestionar y calcular las comisiones de un equipo de vendedores. El programa debe cumplir con los siguientes requisitos:

1. Ingreso de información de vendedores:

- El programa debe permitir agregar varios vendedores.
- Para cada vendedor, el usuario debe ingresar:
 - Nombre del vendedor.
 - Valor de las ventas realizadas (en formato decimal).
 - Tipo de vendedor:
 - 1 para ventas "Puerta a puerta" (con una comisión del 25% sobre las ventas).
 - 2 para ventas por "Telemercadeo" (con una comisión del 15% sobre las ventas).

2. Cálculo de comisiones:

- El programa debe calcular y mostrar la comisión individual de cada vendedor de acuerdo con el tipo de vendedor.
- También debe calcular y mostrar el total de comisiones a pagar entre todos los vendedores.

COMPRAS

3.Opciones del menú:

- El programa debe presentar un menú con las siguientes opciones:
 - 1.Agregar un nuevo vendedor.
 - 2.Generar y mostrar los resultados, incluyendo la comisión individual de cada vendedor y el total de comisiones a pagar.
 - 3.Salir del programa.

4.Validación de entrada:

- El programa debe validar las opciones ingresadas en el menú y manejar errores si el usuario selecciona una opción no válida.

COMPRAS

4. Validación de entrada:

- El programa debe validar las opciones ingresadas en el menú y manejar errores si el usuario selecciona una opción no válida.

Ejemplo de flujo de ejecución:

1. El programa muestra el menú.
2. Si el usuario selecciona la opción "1", se le solicita la información del vendedor (nombre, ventas, y tipo) y se almacena en una lista.
3. Si el usuario selecciona la opción "2", el programa muestra las comisiones de cada vendedor y el total a pagar en comisiones.
4. Si el usuario selecciona la opción "3", el programa finaliza.
5. Si el usuario ingresa una opción inválida, el programa muestra un mensaje de error y vuelve a mostrar el menú.

Menú:

1. Agregar vendedor
2. Generar resultados
3. Salir

Seleccione una opción: 1

Ingrese el nombre del vendedor: Guillermo

Ingrese el valor de las ventas realizadas: 1000000

Ingrese el tipo de vendedor (1=Puerta a puerta, 2=Telemercadeo): 1

Vendedor Guillermo agregado con éxito.

Menú:

1. Agregar vendedor
2. Generar resultados
3. Salir

Seleccione una opción: 2

Comisiones de los vendedores:

Vendedor: Guillermo - Comisión: \$250000.00

Total por pagar en comisiones: \$250000.00

```
# Lista para almacenar los vendedores
lista_vendedores = []

# Ciclo para el menú
while True:
    # Menú de opciones
    print("Menú:")
    print("1. Agregar vendedor")
    print("2. Generar resultados")
    print("3. Salir")

    # Leer la opción seleccionada
    opcion = input("Seleccione una opción: ")

    # Opción 1: Agregar un vendedor
    if opcion == "1":
        nombre = input("Ingrese el nombre del vendedor: ")
        ventas = float(input("Ingrese el valor de las ventas realizadas: "))
        tipo = int(input("Ingrese el tipo de vendedor (1=Puerta a puerta, 2=Telemercadeo): "))

        # Crear un diccionario con la información del vendedor
        vendedor = {
            "nombre": nombre,
            "ventas": ventas,
            "tipo": tipo
        }

        # Agregar el diccionario a la lista de vendedores
        lista_vendedores.append(vendedor)
        print(f"Vendedor {nombre} agregado con éxito.\n")
```

```
# Opción 2: Generar los resultados
elif opcion == "2":
    total_comisiones = 0
    print("\nComisiones de los vendedores:")

    # Recorrer la lista de vendedores
    for vendedor in lista_vendedores:
        # Calcular la comisión según el tipo de vendedor
        if vendedor["tipo"] == 1:
            comision = vendedor["ventas"] * 0.25 # 25% de comisión para Puerta a puerta
        elif vendedor["tipo"] == 2:
            comision = vendedor["ventas"] * 0.15 # 15% de comisión para Telemercadeo
        else:
            print(f"Tipo de vendedor inválido para {vendedor['nombre']}.")
            continue

        # Mostrar la comisión del vendedor
        total_comisiones += comision
        print(f"Vendedor: {vendedor['nombre']} - Comisión: ${comision:.2f}")

    # Mostrar el total por pagar en comisiones
    print(f"\nTotal por pagar en comisiones: ${total_comisiones:.2f}\n")

# Opción 3: Salir
elif opcion == "3":
    print("Saliendo del programa...")
    break # Salir del ciclo y terminar el programa

# Si la opción no es válida
else:
    print("Opción inválida. Intente de nuevo.\n")
```

▶ TALENTO TECH



Universidad
Tecnológica
de Bolívar

www.utb.edu.co/talento-tech