

**Universidad de San Carlos de Guatemala**

**Facultad de Ingeniería Escuela de Ciencias y Sistemas**

**Introducción a la Programación y Computación 2**

## **Practica**

**Catedrático:** Ing. Byron Zepeda

**Auxiliar:** Oscar Roberto Velásquez León

**Sección:** D

**Nombre:** Josué Daniel Rojché García

**Carné:** 201901103

Repositorio:

[https://github.com/DanielGarcia-201901103/IPC2\\_PracticaAuxiliar\\_201901103](https://github.com/DanielGarcia-201901103/IPC2_PracticaAuxiliar_201901103)

## Capturas Resultados

```
=====
----- Menu Principal -----
1. Cargar archivo xml
2. Top 10 de productos con mayor margen de ganancia.
3. Top 10 de productos con mayor valor del inventario.
4. Imprimir datos.
5. Salir.
=====

Ingrese una opcion: 1

Archivo leído exitosamente

=====
----- Menu Principal -----
1. Cargar archivo xml
2. Top 10 de productos con mayor margen de ganancia.
3. Top 10 de productos con mayor valor del inventario.
4. Imprimir datos.
5. Salir.
=====

Ingrese una opcion: 2
```

### Top 10 de productos con mayor margen de ganancia.

\*\*\*\*\*Nivel 1\*\*\*\*\*

Item Code	Margin Level 1	Margin Level 2	Margin Level 3	valor Inventario
B&B3012	1463.46 %	1453.85 %	1159.62 %	\$ 10.4
AGUA1103	508.11 %	494.59 %	413.51 %	\$ 553.52
AGUA1101	386.11 %	381.48 %	321.3 %	\$ 558.36
AGUA1102	311.39 %	305.06 %	241.77 %	\$ 462.94
VLA-26-0025	160.64 %	155.32 %	112.77 %	\$ 35.72
FARMCA7001	150.0 %	135.0 %	100.0 %	\$ 420.0
FARMCA4065	150.0 %	140.0 %	70.0 %	\$ 0.0
GEL-10-0001	143.48 %	134.78 %	126.09 %	\$ 0.0
DIPR1014-1	138.71 %	129.54 %	94.03 %	\$ 1615.76
DIPR1002-1	113.0 %	107.54 %	102.08 %	\$ 842.26

\*\*\*\*\*Nivel 2\*\*\*\*\*

Item Code	Margin Level 1	Margin Level 2	Margin Level 3	valor Inventario
B&B3012	1463.46 %	1453.85 %	1159.62 %	\$ 10.4
AGUA1103	508.11 %	494.59 %	413.51 %	\$ 553.52
AGUA1101	386.11 %	381.48 %	321.3 %	\$ 558.36
AGUA1102	311.39 %	305.06 %	241.77 %	\$ 462.94
VLA-26-0025	160.64 %	155.32 %	112.77 %	\$ 35.72
FARMCA4065	150.0 %	140.0 %	70.0 %	\$ 0.0
FARMCA7001	150.0 %	135.0 %	100.0 %	\$ 420.0
GEL-10-0001	143.48 %	134.78 %	126.09 %	\$ 0.0
DIPR1014-1	138.71 %	129.54 %	94.03 %	\$ 1615.76
DIPR1002-1	113.0 %	107.54 %	102.08 %	\$ 842.26

\*\*\*\*\*Nivel 3\*\*\*\*\*

Item Code	Margin Level 1	Margin Level 2	Margin Level 3	valor Inventario
B&B3012	1463.46 %	1453.85 %	1159.62 %	\$ 10.4
AGUA1103	508.11 %	494.59 %	413.51 %	\$ 553.52
AGUA1101	386.11 %	381.48 %	321.3 %	\$ 558.36
AGUA1102	311.39 %	305.06 %	241.77 %	\$ 462.94
GEL-10-0001	143.48 %	134.78 %	126.09 %	\$ 0.0
VLA-26-0025	160.64 %	155.32 %	112.77 %	\$ 35.72
DIPR1002-1	113.0 %	107.54 %	102.08 %	\$ 842.26
FARMCA7001	150.0 %	135.0 %	100.0 %	\$ 420.0
DIPR1014-1	138.71 %	129.54 %	94.03 %	\$ 1615.76
XTR-10-0023	110.53 %	92.98 %	92.98 %	\$ 0.0

```

=====
----- Menu Principal -----
1. Cargar archivo xml
2. Top 10 de productos con mayor margen de ganancia.
3. Top 10 de productos con mayor valor del inventario.
4. Imprimir datos.
5. Salir.
=====

```

Ingrese una opcion: 3

Top 10 de productos con mayor valor del inventario.

Item Code	Margin Level 1	Margin Level 2	Margin Level 3	valor Inventario
ADRE1002	25.93 %	24.94 %	16.28 %	\$ 42683.52
GLUC2001	12.32 %	12.32 %	10.34 %	\$ 37910.25
ADRE1001	39.43 %	37.96 %	26.71 %	\$ 26490.24
SOGU1001	18.05 %	16.05 %	14.05 %	\$ 24140.34
GOYA3010	16.95 %	9.76 %	7.32 %	\$ 23616.0
NATU1004	30.43 %	26.81 %	17.75 %	\$ 16201.2
GRAJ1002	46.16 %	44.51 %	30.06 %	\$ 15864.1
B&B3005	52.22 %	49.97 %	22.09 %	\$ 14908.96
NATU1005	29.26 %	25.67 %	16.7 %	\$ 14426.3
SOME5001	20.74 %	16.81 %	12.32 %	\$ 13007.77

## Capturas Código

### Código de nodo

 nodo.py
 
 item.py M
  menu.py M

 nodo.py >  Nodo

```

1 class Nodo:
2     def __init__(self, dato):
3         self.dato = dato
4         self.siguiente = None

```

## Código de la lista simplemente enlazada

```
listaSimple.py X menu.py M item.py M [IPC2]PracticaAuxiliar.pdf

listaSimple.py > ListaSimple
1  from nodo import Nodo
2  class ListaSimple:
3      def __init__(self):
4          self.primeros = None
5          self.ultimo = None
6          self.size = 0
7
8      #METODOS PARA AGREGAR DATOS
9      def insertarFinal(self, dato):
10         nuevoNodo = Nodo(dato)
11         self.size +=1
12         #si la lista no tiene ningun dato
13         if self.primeros == None:
14             #el apuntador primeros apunta al nuevo nodo
15             self.primeros = nuevoNodo
16             #el apuntador ultimo apunta al nuevo nodo
17             self.ultimo = nuevoNodo
18         #si la lista ya tiene uno o mas datos se agrega el nuevo nodo
19         else:
20             #el apuntador siguiente apunta al nuevo nodo --->
21             self.ultimo.siguiente = nuevoNodo
22             #el apuntador ultimo apunta al nuevo nodo
23             self.ultimo = nuevoNodo
24
25     def obtenerSize(self):
26         return self.size
27
28     def estaVacia(self):
29         #si el primeros es diferente de nulo No está vacia
30         if self.primeros !=None:
31             return True
32         #Si el primeros es igual a nulo Si está vacia
33         if self.primeros ==None:
34             return False
```

```

35
36 ✓ def imprimir(self):
37     contador = 0
38     nodoTemporal = Nodo("")
39     nodoTemporal = self.primerO
40 ✓ while nodoTemporal != None:
41     contador +=1
42     print("ItemCode: ",nodoTemporal.dato.item)
43     print("QuantityOnHand: ",nodoTemporal.dato.quantityOnHand)
44     print("PriceLevel1: $",nodoTemporal.dato.priceLevel1)
45     print("PriceLevel2: $",nodoTemporal.dato.priceLevel2)
46     print("PriceLevel3: $",nodoTemporal.dato.priceLevel3)
47     print("LastTotalUnitCost: $",nodoTemporal.dato.lastTotalUnitCost)
48     print("MarginLevel1: ",nodoTemporal.dato.marginLevel1,"%")
49     print("MarginLevel2: ",nodoTemporal.dato.marginLevel2,"%")
50     print("MarginLevel3: ",nodoTemporal.dato.marginLevel3,"%")
51     print("valorInventario: $",nodoTemporal.dato.valorInventario)
52     print()
53     nodoTemporal = nodoTemporal.siguiente
54

```

```

54
55 def imprimirTop(self):
56     print('-'*96)
57     print("| {:<16} | {:<16} | {:<16} | {:<16} | {:<16} |".format('Item Code','Margin Level 1',
58     'Margin Level 2', 'Margin Level 3',
59     'valor Inventario'))
60     print('-'*96)
61     contador = 0
62     nodoTemporal = Nodo("")
63     nodoTemporal = self.primerO
64     while nodoTemporal != None:
65         contador +=1
66         if contador <= 10:
67             print("| {:<16} | {:<16} | {:<16} | {:<16} | {:<16} |".format(str(nodoTemporal.dato.item),
68             str(nodoTemporal.dato.marginLevel1)+str(" %"),
69             str(nodoTemporal.dato.marginLevel2)+str(" %"),
70             str(nodoTemporal.dato.marginLevel3)+str(" %"),
71             str("$ ") +str(nodoTemporal.dato.valorInventario)))
72     print()
73     nodoTemporal = nodoTemporal.siguiente

```

## Código de ordenamiento burbuja

```
74
75  def ordenamientoBurbuja(self, llave):
76      #las variables hacen referencia al primer elemento de la lista
77      actual = self.primeros
78      aux = self.primeros
79      #si la lista está vacía agrega el dato a la lista
80      if actual.siguiente != None and aux != None:
81          #obtiene el primer dato de la lista
82          i = self.primeros
83          while i != None:
84              #obtiene el dato siguiente de la lista
85              j = i.siguiente
86              while j != None:
87                  #compara los datos para saber cual es el mayor
88                  if llave == "valor inventario":
89                      i1 = i.dato.valorInventario
90                      j1 = j.dato.valorInventario
91                  elif llave == "margen level 1":
92                      i1 = i.dato.marginLevel1
93                      j1 = j.dato.marginLevel1
94                  elif llave == "margen level 2":
95                      i1 = i.dato.marginLevel2
96                      j1 = j.dato.marginLevel2
97                  elif llave == "margen level 3":
98                      i1 = i.dato.marginLevel3
99                      j1 = j.dato.marginLevel3
100
101                  if i1 < j1:
102                      #cambia el orden de los datos
103                      temporal = i.dato
104                      i.dato = j.dato
105                      j.dato = temporal
106                      #pasa al siguiente dato de la lista
107                      j = j.siguiente
108                      #pasa al siguiente dato de la lista
109                      i = i.siguiente
```

## Código para realizar los cálculos

```
menu.py M  item.py M X  [IPC2]PracticaAuxiliar.pdf
item.py > Item
1  class Item():
2      item: str
3      quantityOnHand: float
4      priceLevel1: float
5      priceLevel2: float
6      priceLevel3: float
7      lastTotalUnitCost: float
8      marginLevel1: float
9      marginLevel2: float
10     marginLevel3: float
11     valorInventario : float
12
13     def __init__(self, item, quantityOnHand, priceLevel1, priceLevel2, priceLevel3, lastTotalUnitCost):
14         self.item = item
15         self.quantityOnHand = quantityOnHand
16         self.priceLevel1 = priceLevel1
17         self.priceLevel2 = priceLevel2
18         self.priceLevel3 = priceLevel3
19         self.lastTotalUnitCost = lastTotalUnitCost
20         self.marginLevel1 = self.__calcularMargenes(self.priceLevel1, self.lastTotalUnitCost)
21         self.marginLevel2 = self.__calcularMargenes(self.priceLevel2, self.lastTotalUnitCost)
22         self.marginLevel3 = self.__calcularMargenes(self.priceLevel3, self.lastTotalUnitCost)
23         self.valorInventario = self.__calcularValorInventario()
24
25     def __calcularMargenes(self, precio, costo):
26         if costo != 0:
27             margen = ((precio - costo)/costo) * 100
28             resultado = round(margen, 2)
29             return resultado
30
31     def __calcularValorInventario(self):
32         valor = self.lastTotalUnitCost * self.quantityOnHand
33         resultado = round(valor, 2)
34         return resultado
```