

Evaluación Primer Corte

8 de marzo de 2022

1. La Figura 1, presenta la información que se requiere representar en memoria, con el fin de desarrollar las funcionalidades del software que gestiona las cuentas de unas mesas en un café.

Mesa 1	
Producto	Precio
Café	2000
Helado	3000
Agua con gas	2300
Mesa 2	
Producto	Precio
Papas gratinadas	8000
Banan split	7000
Mesa 3	
Producto	Precio
Té helado	2560
Tortilla huevo	5300
Café	2000

Figura 1: Información de lo que se está consumiendo en cada mesa

Resolver entonces los siguientes requerimientos:

- (a) (*Valor 0.5*) Proponer la estructura de datos que considere adecuada para cargar la información de las mesas en memoria, empleando las estructuras de datos nativas de Python. Justificar la estructura de datos elegida.
 - (b) (*Valor 1.0*) Utilizar la estructura de datos propuesta en el numeral anterior para mostrar cómo se cargaría en el código la información presentada en la Figura 1.
 - (c) (*Valor 1.0*) Escribir una rutina para totalizar cada una de las mesas en una estructura de datos como la propuesta, pero que podría estar variando tanto en número de mesas, como en cantidad de productos.
 - (d) (*Valor 1.0*) Se requiere una funcionalidad que permita al usuario ingresar el número o nombre de la mesa (dependiendo de la estructura de datos que se haya planteado), y retornar el último producto que ha sido adicionado al pedido de la mesa ingresada.
 - (e) (*Valor 0.5*) Escribir un procedimiento que muestre en pantalla el producto más costoso que se está consumiendo, junto con su precio y la mesa en la cual ha sido ordenado.
2. (*Valor 1.0*) Observar el Código 1, donde se especifican dos estructuras de datos para representar en memoria una red de puntos. Se requiere escribir una funcionalidad para encontrar la distancia entre un nodo origen y un nodo destino ingresados por el usuario. Nótese que el usuario ingresará las etiquetas para seleccionar los nodos.

```
import pprint as pp
#Estructura de datos para los nodos
nodos = []
nodos.append([8,11])#A
nodos.append([10,10])#E
nodos.append([7,8])#B
nodos.append([9,7])#D
nodos.append([9.5,5])#C
#Estructura de datos para las etiquetas
tabla_asociativa = ['A','E','B','D','C']
nodoOrigen = input("Ingrese la etiqueta del nodo origen: ")
nodoDestino = input("Ingrese la etiqueta del nodo destino: ")
#Funcionalidad para encontrar la distancia entre nodo origen y nodo destino
```

Código 1: Contenedores y entradas establecidas para el requerimiento