

Introducción a la Programación

Tercer Entregable de laboratorio

Ejercicio 3

En una tienda de comestibles en línea, se desea implementar un sistema de procesamiento de pedidos. Los clientes realizan sus pedidos en línea y los pedidos se procesan en el orden en que se reciben. Cada pedido se representa como un diccionario con la siguiente estructura:

```
pedido = { 'id': 1, 'cliente': 'Juan', 'productos': {'Manzana': 2, 'Pan': 4, 'Factura': 6} }
```

Notar que 'productos' es a su vez un diccionario donde la clave es el nombre del producto, y su valor representa la cantidad de ese producto a solicitar. En el ejemplo: el cliente 'Juan' cuyo 'id' es 1, está solicitando 2 unidades de 'Manzana', 4 unidades de 'Pan' y 6 unidades de 'Factura'.

El procesamiento de pedidos se realiza de la siguiente manera:

- Se toma el primer pedido de la cola.
- Se verifica el stock de cada producto en el pedido. Se descuenta del stock disponible los productos solicitados en un pedido.
- Se calcula el precio total del pedido.
- Se marca el pedido como 'completo' o 'incompleto' dependiendo si había stock suficiente o no para **todos** los productos. Es decir, si no había stock suficiente de algún producto, el estado será 'incompleto' aunque se procesarán todos los productos.
- Se pasa al siguiente pedido en la cola y se repiten los pasos anteriores hasta que no queden más pedidos en la cola.

El stock y los precios de los productos se obtienen de diccionarios con la siguiente estructura:

```
stock_productos = { "Manzana": 10, "Leche": 5, "Pan": 3, "Factura": 0 }  
precio_productos = { "Manzana": 3.5, "Leche": 5.5, "Pan": 3.5, "Factura": 2.75 }
```

Se pide implementar la función `procesar_pedidos(in pedidos: Queue[Dict[str, Union[int, str, Dict[str, int]]], inout stock_productos: Dict[str, int], in precios_productos: Dict[str, float]) -> List[Dict[str, Union[int, str, float, Dict[str, int]]]` que retorne una lista de diccionarios con los pedidos procesados.

Por ejemplo, para el siguiente input de 'pedidos':

```
pedidos = [{  
    'id': 21,  
    'cliente': 'Gabriela',  
    'productos': {'Manzana': 2}  
}, {  
    'id': 1,  
    'cliente': 'Juan',  
    'productos': {'Manzana': 2, 'Pan': 4, 'Factura': 6}  
}]
```

y los inputs 'stock_productos' y 'precio_productos' que figuran anteriormente, el resultado esperado es:

```
pedido_procesado = [{  
    'id': 21,  
    'cliente': 'Gabriela',  
    'productos': {'Manzana': 2}
```

```

    'precio_total': 7.0,
    'estado': 'completo'
  }, {
    'id': 1,
    'cliente': 'Juan',
    'productos': {'Manzana': 2, 'Pan': 3, 'Factura': 0}
    'precio_total': 17.5,
    'estado': 'incompleto'
  }]

```

Aclaraciones:

- pedido: Dict[str, Union[int, str, Dict[str, int]]]: Significa que el tipo de dato de 'pedido' es un diccionario donde su clave es de tipo *str* y su valor de algunos de los siguientes tipos: *int*, *str* ó *Dict[str, int]*. Es decir, puede haber valores de tipo *int*, otros de tipo *str* y/o otros de tipo *Dict[str, int]*. Sin embargo, esta notación no es soportada en el CMS, por lo tanto, verán en el template que descargan que su anotación es simplemente 'pedidos: Queue'.
- Se puede asumir que todos los productos en 'pedidos' existen tanto en 'stock_productos' (pueden tener cantidad 0) como en 'precio_productos'.