**¿Como funciona el programa?**

El funciona funciona principalmente con funciones valga la redundancia. Varias con distintos propositos pero las más importantes para el programa son las que hacen uso del manejo de claves de diccionarios, por ejemplo:

def buscar\_factura(id\_factura): #Se pone como parametro el id de factura

for factura in facturas: #Se recorre entre la lista de facturas

if factura['id\_factura'] == id\_factura:

return factura

return None

Cabe aclarar en el anterior fragmento se pone una condición, en la cual si el argumento ('id\_factura') que puso el usuario coincide con el id de una factura en la lista de facturas, se retorna la factura, de lo contrario retorna None.

¿Y en qué nos sirve eso?

Por ejemplo para condicionar otras funciones del programa y resolver la siguiente pregunta:

¿Cómo sabe el programa qué factura en especifico quiero manipular?

Pues sería por ejemplo así:

def agregar\_producto\_a\_factura():

id\_factura = int(input("ID de la factura: "))

factura = buscar\_factura(id\_factura)

if factura:

(bloque de codigo si existe una factura con la ID introducida)

else:

print("No existe una factura con ese ID")

Se pone una condición en la que si la factura existe con el ID introducido se ejecuta un bloque de código (es importante saber que cualquier retorno en variable o una lista con elementos dará el valor de True) y si no existe se ejecuta el else (al retornar el None, osea, un valor False)

Esa función la ocupamos dentro de otras funciones que manipulan facturas especificas (como agregar productos a factura o mostrar factura)

**¿Qué variables ocupados?**

Ocupamos 3 variables globales

productos = {} un diccionario que guarda el nombre de cada producto como clave y su valor es otro diccionario

(que contiene dos llaves y dos valores)

ejemplo:

productos = {"Leche": {"nombre": "Leche", "precio": 2.0}}:

facturas = [] otro diccionario que guarda más diccionarios

ejemplo:

producto=[{"id\_factura": 2, "cliente": "Bayron", "items": [{"nombre": "Leche", "precio": 2.0}, 1], "total": 2.0}]

next\_id\_factura = 1 Es un entero que estará incrementandose constantemente por cada factura creada

**Ciclos ocupados en el programa.**

Validación de entradas:

En el código hay varios ciclos, los cuales son para que se repita un input si el usuario no ha ingresado un tipo de dato correcto (por ejemplo, cuando se le pide poner una cantidad de productos en la factura e introduce una letra (lo que se le está pidiendo es un número entero, y por ende no tendría sentido lo de la letra)

Texto

Descripción generada automáticamente

En el ejemplo se puede ver como si es un número entero positivo, el ciclo se romperá con break y se dejará de pedir que el usuario introduzca un dato con el input de cantidad. Si introduce una menor a 0 le dirá que tiene que ser positivo. Y si introduce un float o letra le saldrá la excepción de error de valor.