

1. **Sub-Competencia:** Separa el código en funciones pequeñas reusables, haciendo uso correcto de paso por parámetros y return (avance 3).

Error original: El código está revuelto y no tenía el orden adecuado para que este fuera funcional y pudiera funcionar de una manera correcta.

```
1 + products = [  
2 + {  
3 +     "name": "Huevo",  
4 +     "price": 15.50,  
5 +     "store": "Walmart Bernardo Quintana"  
6 + },  
7 + {  
8 +     "name": "Aceite",  
9 +     "price": 22.50,  
10 +     "store": "Walmart Bernardo Quintana"  
11 + },  
12 + {  
13 +     "name": "Tortilla",  
14 +     "price": 32.50,  
15 +     "store": "Walmart Bernardo Quintana"  
16 + },  
17 + ]  
18 + def calculate_basic_basket():  
19 +     total=0  
20 +     for product in products:  
21 +         total+=product["price"]  
22 +     return total  
23 + def average_price():  
24 +     average = calculate_basic_basket()/len(products)  
25 +     return average  
26 +  
27 + def valida (num):  
28 +     return ((num>1) and (num<=4))  
29 +  
30 + opcion = int (input("INTRODUCE LA OPCIÓN: "))  
31 +  
32 + if valida(opcion):  
33 +  
34 +     if opcion == 1:  
35 +         value=calculate_basic_basket()  
36 +         print(f"El valor total de la canasta básica es : {value}")  
37 +  
38 +     elif opcion == 2:  
39 +         average=average_price()  
40 +         print (f"El precio promedio de la canasta básica es : {average}")  
41 +
```

Cambio realizado: El código está limpio y se pueden observar códigos reusables que más de una vez pueden ser llamados y utilizados.

Líneas de código donde se ve la corrección:

```
29  
30 def av_prod_price(listaproducto): # precio prom de cada producto  
31     a = 0  
32     for el in range(0, len(listaproducto)):  
33         c = listaproducto[el]  
34         priceprod = c["price"]  
35         a = a + priceprod  
36     p = a / (len(listaproducto))  
37     return p  
38  
39  
40 def average_price(prod_list, lista):  
41     acbp = 0  
42     for j in range(0, len(prod_list)): # n de prod  
43         clave = lista[j]  
44         listaproducto = prod_list[clave]  
45         prom = av_prod_price(listaproducto)  
46         acbp = prom  
47     return acbp  
48  
49  
50 def valida(num):  
51     return ((num >= 1) and (num <= 4))  
52  
53  
54 products_huevo = [  
55     {  
56         "name": "Huevo",  
57         "price": 30.23,  
58         "store": "Walmart Bernardo Quintana"  
59     },  
60     {  
61         "name": "Huevo",  
62         "price": 40.90,  
63         "store": "HEB Juriquilla"  
64     },  
65     {  
66         "name": "Huevo",  
67         "price": 31.88,  
68         "store": "Chedraui Selecto Juriquilla"
```

Líneas 1-52 podemos observar las funciones

Líneas 56-152 Observamos los diccionarios ordenados

Líneas 158-172 definiciones de lista

Líneas 173-188 se trata del menú y de las opciones que se van a dar en el menú

2. Sub-Competencia: componente: aplica estructuras condicionales para resolver un problema (avance 4)

Error original: Las opciones agregadas en los if se realizaban no se utilizaban elif ni una función donde se comprobará la correcta composición de los elementos indicados por el usuario.

if opcion == 1:

```
value=calculate_basic_basket()
```

```
print(f" El valor total de la canasta básica es : {value}")
```

if opcion == 2:

```
average1=average_price()
```

```
print (f"El precio promedio de la canasta básica es : {average1}")
```

Cambio realizado: Se agregó una función en la que se compruebe la opción agregada por el usuario, además que se agregaron ifelse para un mejor funcionamiento del programa

Líneas de código donde se ve la corrección:

```
def valida(num):  
    return ((num >= 1) and (num <= 4))  
  
174 if valida(opcion):  
175     if opcion == 1:  
176         show_products(product_names)  
177         option = int(input("Ingrese el número del producto que desea saber el menor costo"))  
178         p_name = product_names[option - 1]  
179         min_p = precio_menor(products, p_name)  
180         print(f"El mejor lugar para adquirir el producto es: {min_p['store']}, y su precio es de: ${min_p['price']}")  
181  
182     elif opcion == 2:  
183         value = calculate_basic_basket(products, product_names)  
184         print(f" El valor total de la canasta básica es de: ${value}")  
185  
186     elif opcion == 3:  
187         average1 = average_price(products, product_names)  
188         print(f"El precio promedio de todos los productos de la canasta básica es de: ${average1}")  
189
```

50 y 51 función declarada para la correcta función del programa.

174-189 se agregaron elif's y algunas otras cosas para la buena función