

# Ciclo 1

## Fundamentos de programación en Python

# Sesión 25: Sesión Preparación Reto 5

Programa Ciencias de la Computación e Inteligencia  
Artificial

Escuela de Ciencias Exactas e Ingeniería  
Universidad Sergio Arboleda  
Bogotá

# Agenda

1. Enunciado del reto
2. Ingreso de datos para el reto
3. Comparación de parámetros
4. Pasos a considerar en la solución del reto
5. Ejercicios

# 1. Enunciado del reto

Una tienda vende diferentes productos, usualmente frutas, dulces y algunos tipos de carne. Con el propósito de mejorar el control sobre las ventas y el inventario de la tienda, el gerente decide construir una aplicación que le permita almacenar la información de los productos y realizar algunos cálculos sobre los datos.

En la primera parte del reto se debe construir una base de datos que almacene la información de los productos disponibles en la tienda. La base de datos será representada mediante un diccionario de Python llamado productos que tendrá por llave el código del producto y por valor una lista formada por los atributos: nombre, precio e inventario. La Tabla 1 presenta los productos disponibles a la fecha.

# 1. Enunciado del reto

código	nombre	precio	inventario
1	Manzanas	5000.0	25
2	Limonas	2300.0	15
3	Peras	2700.0	33
4	Arandanos	9300.0	5
5	Tomates	2100.0	42
6	Fresas	4100.0	3
7	Helado	4500.0	41
8	Galletas	500.0	8
9	Chocolates	3500.0	80
10	Jamon	15000.0	10

Tabla 1: Productos disponibles en la tienda

# 1. Enunciado del reto

Es necesario construir 3 funciones que representen las operaciones de: AGREGAR, ACTUALIZAR y BORRAR los productos disponibles. Se debe implementar una función independiente por cada una de las acciones mencionadas. En este caso, para poder realizar las operaciones de ACTUALIZAR o BORRAR es necesario especificar todos los atributos del producto.

# 1. Enunciado del reto

Adicionalmente, se está interesado en analizar los datos de los productos disponibles y conocer: el nombre del producto con el precio mayor; el nombre del producto con el precio menor; el promedio de precios de todos los productos y el valor total del inventario a la fecha. Este último se obtiene multiplicando el precio de cada producto por el inventario disponible y luego sumando todos los resultados. Por ejemplo, al calcular estos 4 valores para los datos disponibles en la Tabla 1 obtendríamos:

- Producto precio mayor: Jamon
- Producto precio menor: Galletas
- Promedio de precios: 4900.0
- Valor del inventario: 101410.0

# 1. Enunciado del reto

Entrada	<p>Cada uno de los casos de prueba estará compuesto por dos líneas.</p> <ul style="list-style-type: none"><li>La primera línea estará formada por una cadena de texto que identifica la operación a realizar. En este caso, las operaciones validas son:<ul style="list-style-type: none"><li>ACTUALIZAR</li><li>BORRAR</li><li>AGREGAR</li></ul></li><li>La segunda línea estará formada por 4 valores (código, nombre, precio, inventario) que representan el producto sobre el cual se quiere realizar la operación.</li><li>En el caso de la operación ACTUALIZAR la segunda línea debe contener el código y los nuevos valores del producto</li><li>En el caso de la operación BORRAR se deben especificar todos los atributos del producto a eliminar</li></ul>
Salida	<p>La salida estará representada por una única línea formada por cuatro valores:</p> <ul style="list-style-type: none"><li>Nombre del producto con el precio mayor</li><li>Nombre del producto con el precio menor</li><li>Promedio de precios</li><li>Valor del inventario</li></ul> <ul style="list-style-type: none"><li>Estos 4 valores deben imprimirse después de realizar las operaciones solicitadas en la entrada de datos.</li><li>Los valores numéricos deben imprimirse con un número decimal</li><li>En caso de solicitar ACTUALIZAR o BORRAR un producto que no existe (es decir, que el código del producto no se encuentra en la base de datos), se debe imprimir "ERROR"</li><li>En caso de solicitar AGREGAR un producto cuyo código ya existe en la base de datos se debe imprimir "ERROR"</li></ul>



## 2. Ingreso de datos para el reto

La calificación del reto es automático. Se debe considerar la forma de ingresar los datos al calificador automático. En esta oportunidad el uso de funciones en Python ayuda en este proceso. La entrada de datos son dos líneas con dos parámetros. La Primera línea es el nombre de la función a realizar (AGREGAR, ACTUALIZAR, BORRAR). La segunda línea los datos del producto. Se sugiere la siguiente forma:

```
def leer_datos():  
    operacion = input()  
    producto = input().split()  
    producto[0] = int(producto[0])  
    producto[2] = float(producto[2])  
    producto[3] = int(producto[3])  
    return operacion, producto
```

### 3. Crear la base de datos.

Para hacer el computo de los productos, hay que ingresar la base de datos en un diccionario. Se sigue la siguiente estructura:

```
▶ database = {  
    1: ['Manzanas', 5000.0, 25],  
    2: ['Limones', 2300.0, 15],  
    3: ['Peras', 2700.0, 33],  
    4: ['Arandanos', 9300.0, 5],  
    5: ['Tomates', 2100.0, 42],  
    6: ['Fresas', 4100.0, 3],  
    7: ['Helado', 4500.0, 41],  
    8: ['Galletas', 500.0, 8],  
    9: ['Chocolates', 3500.0, 80],  
    10: ['Jamon', 15000.0, 10]  
}
```

## 4. Crear de funciones.

Las funciones claves son tres: AGERGAR, ACTUALIZAR y BORARA. Estas funciones actúan sobre el diccionario que contiene los valores de la tienda. A continuación, las estructuras que se siguen de las funciones claves:

```
[ ] def borrar(database, producto):  
    if producto[0] in database:  
        database.pop(producto[0])  
        return True  
    return False
```

```
[ ] def agregar(database, producto):  
    if producto[0] in database:  
        return False  
    index = producto[0]  
    producto.pop(0)  
    database[index] = producto  
    return True
```

```
[ ] def actualizar(database, producto):  
    if producto[0] in database:  
        index = producto[0]  
        producto.pop(0)  
        database[index] = producto  
        return True  
    return False
```

## 4. Crear de funciones.

Hay operaciones de respaldo que se deben diseñar, por ejemplo: calculo del precio mayor, el precio menor, el promedio de precios, el valor del inventario. A continuación se proponen una forma de crear estas funciones

```
def precio_mayor(database):  
    mayor = list(database.keys())[0]  
    for i in database:  
        if database[i][1] > database[mayor][1]:  
            mayor = i  
    return database[mayor][0]
```

```
def promedio_precios(database):  
    promedio = 0  
    for i in database:  
        promedio += database[i][1]  
    promedio /= len(database)  
    return promedio
```

```
def precio_menor(database):  
    menor = list(database.keys())[0]  
    for i in database:  
        if database[i][1] < database[menor][1]:  
            menor = i  
    return database[menor][0]
```

```
def valor_inventario(database):  
    valor_inventario = 0.0  
    for i in database:  
        valor_inventario += database[i][1] * database[i][2]  
    return valor_inventario
```

## 5. Crear un main().

Una vez teniendo las funciones principales y las funciones de respaldo, tener una función *main()*, en donde se puedan hacer el llamado a esas funciones es una buena practica. Por otro lado, hay que considerar la forma en la cual llama las funciones clase.

A continuación, se sugiere una posible forma de hacer la llamada a estas funciones:

```
operacion, producto = leer_datos()

if operacion == 'AGREGAR':
    flag = agregar(database, producto)
elif operacion == 'BORRAR':
    flag = borrar(database, producto)
elif operacion == 'ACTUALIZAR':
    flag = actualizar(database, producto)
```

## 6. Pasos a considerar para la solución del reto

1. Entender el problema: realizar un diagrama que describa el problema
2. Plantear la solución: realizar un diagrama con las operaciones a realizar
3. Ingresar datos: hacer pruebas ingresando datos teniendo que la entrada son dos líneas. La primera línea debe tener la función a realizar y la segundo la descripción del producto.
4. Crear la base de datos como un diccionario
5. Crear las funciones claves y las funciones de apoyo
6. Crear la función principal donde se hace el llamado a las funciones
7. Imprimir el resultado



## 7. Ejercicios

Crear una una matriz de  $m \times n$  elementos la calificación obtenida por  $m$  estudiantes (a los que conocemos por su número de lista) en la evaluación de  $n$  ejercicios entregados semanalmente (cuando un ejercicio no se ha entregado, la calificación es  $-1$ ). Diseñar funciones que efectúen los siguiente cálculos:

- Dado el número de un alumno, devolver el número de ejercicios entregados.
- Dado el número de un alumno, devolver la media sobre los ejercicios entregados.
- Dado el número de un alumno, devolver la media sobre los ejercicios entregados si los entregó todos; en caso contrario, la media es 0.
- Devolver el número de todos los alumnos que han entregado todos los ejercicios y tienen una media superior a 3.5 puntos.

## 7. Ejercicios

- Dado el número de un ejercicio, devolver la nota media obtenida por los estudiantes que lo presentaron. Dado el número de un ejercicio, devolver la nota más alta obtenida.
- Dado el número de un ejercicio, devolver la nota más baja obtenida.
- Dado el número de un ejercicio, devolver el número de estudiantes que lo han presentado.
- Devolver el número de abandonos en función de la semana. Consideramos que un alumno abandonó en la semana x si no ha entregado ningún ejercicio desde entonces. Este procedimiento mostrará en pantalla el número de abandonos para cada semana (si un alumno no ha entregado nunca ningún ejercicio, abandonó en la semana cero).



# Preguntas

