

Cargo:	Docente		
Nombre:	M.Sc. David Fabián Cevallos Salas		
Asignatura:	Administración de Base de Datos		
Carrera:	Marketing Digital y Comercio Electrónico	Nivel:	Quinto nivel
Estudiante:			

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

APLICACIÓN DE TÉCNICAS DE FILTRADO Y ACTUALIZACIÓN EN UNA BASE DE DATOS NO RELACIONAL.

1. Objetivos

- Aprender a trabajar con arreglos JSON en Python: El estudiante aprenderá a filtrar y actualizar datos en un arreglo JSON de productos usando técnicas como el slicing de arreglos y la compresión de listas.
- Aplicar operaciones de filtrado: Usar las herramientas de Python para dividir un arreglo JSON en subconjuntos más pequeños basados en condiciones específicas.
- Realizar actualizaciones de datos: Modificar campos dentro de un objeto JSON para simular operaciones de actualización en una base de datos o estructura similar.

2. Antecedentes/Escenario

Imagina que trabajas en un carrito de compras de una tienda en línea. Este carrito contiene una lista de productos, donde cada producto es representado por un objeto JSON con detalles como nombre, precio, cantidad y categoría. Como parte de una actividad práctica, se te pide que realices dos operaciones principales:

1. **Filtrado de productos:** Filtrar los productos basándote en ciertas condiciones, como categoría o precio, y organizar los productos en listas más pequeñas para análisis o presentación.
2. **Actualización de productos:** Actualizar ciertos atributos de los productos, como el precio o la cantidad, cuando sea necesario.

3. Recursos necesarios

- **Google Colab:** Para ejecutar el código Python en la nube.



- **Conocimientos Básicos en Python:** Familiaridad con listas, diccionarios y operaciones básicas de programación.

4. Planteamiento del problema

Se te dará un arreglo JSON de productos en el carrito de compras, y deberás realizar las siguientes tareas:

1. Filtrar los productos según su **categoría** y **precio**.
2. Actualizar el **precio** de algunos productos y modificar sus **cantidades** según ciertas condiciones.

Tu tarea es organizar el código para realizar el filtrado y la actualización de manera eficiente usando Python.

5. Pasos por realizar

5.1. Paso 1: Filtrado de Productos

- **Filtrar productos por categoría:** Usando **slicing** y **compresión de listas**, separa los productos en categorías específicas. Por ejemplo, filtra todos los productos de la categoría "ropa".

```
productos = [  
    {"nombre": "Camiseta", "precio": 20, "cantidad": 2, "categoria": "ropa"},  
    {"nombre": "Zapatos", "precio": 50, "cantidad": 1, "categoria": "calzado"},  
    {"nombre": "Gorra", "precio": 15, "cantidad": 3, "categoria": "accesorios"},  
    {"nombre": "Pantalón", "precio": 30, "cantidad": 5, "categoria": "ropa"},  
    {"nombre": "Botines", "precio": 80, "cantidad": 1, "categoria": "calzado"}  
]  
  
# Filtrar productos de la categoría "ropa"  
productos_ropa = [producto for producto in productos if producto["categoria"] == "ropa"]  
print(productos_ropa)
```

5.2. Paso 2: Actualización de Datos

- **Actualizar el precio de un producto:** Realiza una operación de actualización para modificar el precio de los productos. Por ejemplo, aumentar el precio de todos los productos de "ropa" en un 10%.



```
# Actualizar el precio de productos de la categoría "ropa"
for producto in productos:
    if producto["categoria"] == "ropa":
        producto["precio"] *= 1.10 # Aumenta el precio en un 10%
print(productos)
```

6. Desarrollo

