

# Ejercicios de aplicación 🧐

## ESTRUCTURAS DE DATO EN PYTHON Y SENTENCIAS ITERATIVAS

Es hora de que pongas en práctica todo lo aprendido. 🧐

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes desafíos que podrás resolver de forma independiente y a tu ritmo.

Más adelante conseguirás las resoluciones para que valides tus respuestas y puedas monitorear tu progreso. 😊

### ¡Manos a la obra!

---

#### Desafío 🎯

Implementar un sistema de gestión de inventario utilizando estructuras de datos en Python. El desafío incluye:

- Crear un diccionario que represente un inventario de productos, donde cada clave sea el nombre del producto y su valor un diccionario con detalles como **cantidad**, **precio**, y **categoría**.
- Escribir funciones que permitan:
  - **Agregar productos** al inventario.
  - **Eliminar productos** del inventario.
  - **Actualizar la cantidad y precio** de productos.
  - **Listar los productos** disponibles en una categoría específica.
  - **Calcular el valor total del inventario** sumando el valor de todos los productos (cantidad \* precio).

#### Paso a paso:

1. Crear el diccionario principal **inventario**.
  2. Desarrollar las funciones **agregar\_producto()**, **eliminar\_producto()**, **actualizar\_producto()**, **listar\_por\_categoria()**, y **calcular\_valor\_total()**.
  3. Probar cada función en diferentes escenarios, asegurándote de que manejan correctamente las operaciones de inventario.
- 


#### ¿Dónde se lleva a cabo? 🏠

Visual Studio Code o cualquier editor de Python de tu preferencia.

---

# Ejercicios de aplicación

## ESTRUCTURAS DE DATO EN PYTHON Y SENTENCIAS ITERATIVAS

Tiempo de dedicación 

1:30 horas

---

Recursos 

Material de referencia sobre estructuras de datos en Python, video sobre manipulación de diccionarios y listas (opcional).

---

Plus 

Si quieres un desafío extra, implementa una función que permita **exportar el inventario a un archivo CSV** para facilitar el manejo externo de datos.

---

 Condición

Esta práctica o ejercitación **no requiere ser entregada** y/o evaluada por el mentor. No obstante, puedes compartir tus resultados con el resto de los participantes y construir conocimiento en conjunto.

---

### Resolución del ejercicio:

#### Código Completo y Explicación

```
import csv # Necesario para la exportación a CSV

# Diccionario inicial de inventario

inventario = {}

# Función para agregar productos al inventario
```

```
def agregar_producto(nombre, cantidad, precio, categoria):

    inventario[nombre] = {'cantidad': cantidad, 'precio':
precio, 'categoria': categoria}

# Función para eliminar productos del inventario

def eliminar_producto(nombre):

    if nombre in inventario:

        del inventario[nombre]

    else:

        print("El producto no existe en el inventario.")

# Función para actualizar cantidad y precio de productos

def actualizar_producto(nombre, cantidad=None, precio=None):

    if nombre in inventario:

        if cantidad is not None:

            inventario[nombre]['cantidad'] = cantidad

        if precio is not None:

            inventario[nombre]['precio'] = precio

    else:

        print("El producto no existe en el inventario.")

# Función para listar productos de una categoría específica

def listar_por_categoria(categoria):
```

```
for nombre, detalles in inventario.items():

    if detalles['categoria'] == categoria:

        print(f"Producto: {nombre}, Cantidad:
{detalles['cantidad']}, Precio: {detalles['precio']}")

# Función para calcular el valor total del inventario

def calcular_valor_total():

    valor_total = sum(detalles['cantidad'] *
detalles['precio'] for detalles in inventario.values())

    return valor_total

# Función para exportar el inventario a un archivo CSV

def exportar_inventario_csv(nombre_archivo):

    with open(nombre_archivo, 'w', newline='') as archivo_csv:

        campos = ['Producto', 'Cantidad', 'Precio',
'Categoria']

        writer = csv.writer(archivo_csv)

        writer.writerow(campos)

        for nombre, detalles in inventario.items():

            writer.writerow([nombre, detalles['cantidad'],
detalles['precio'], detalles['categoria']])

# Ejemplo de uso de las funciones

agregar_producto("Laptop", 10, 700, "Electrónica")
```

# Ejercicios de aplicación 🧐

## ESTRUCTURAS DE DATO EN PYTHON Y SENTENCIAS ITERATIVAS

```
agregar_producto("Manzana", 50, 0.5, "Alimentos")

listar_por_categoria("Electrónica")

print("Valor total del inventario:", calcular_valor_total())

exportar_inventario_csv("inventario.csv")
```

1. **Agregar Productos:** Usa `agregar_producto()` para añadir productos con detalles.
2. **Eliminar Productos:** Usa `eliminar_producto()` para borrar un producto.
3. **Actualizar Productos:** `actualizar_producto()` cambia `cantidad` y `precio`.
4. **Listar por Categoría:** Filtra y muestra productos de una categoría.
5. **Valor Total:** `calcular_valor_total()` suma el valor de todo el inventario.
6. **Exportar a CSV:** `exportar_inventario_csv()` guarda el inventario en un archivo CSV.

---

Este ejercicio te permitirá desarrollar y afianzar el uso de estructuras de datos como diccionarios en Python, así como el uso de funciones iterativas y condicionales.