

## Archivos - Práctica

### Objetivo:

Comprender y aplicar el uso de archivos de texto en Python, desarrollando un pequeño programa que lea, procese y almacene información sobre productos de manera persistente. Se busca que el estudiante manipule datos a través de estructuras como listas y diccionarios, integrando lectura, escritura y buenas prácticas con archivos.

### Resultados de aprendizaje:

#### Lectura y escritura de archivos:

El estudiante desarrollará la habilidad de leer y escribir información en archivos de texto mediante ejemplos prácticos, reconociendo los modos `'r'`, `'w'` y `'a'`.

**Manejo de estructuras de datos:** Será capaz de convertir líneas de texto en listas y diccionarios, y manipular esta información en memoria de forma eficiente.

**Persistencia y reutilización de datos:** Entenderá el concepto de persistencia de datos y será capaz de actualizar archivos sin borrar la información previa, reutilizando y modificando registros.

**Buenas prácticas al trabajar con archivos:** Aplicará el uso de `with open()` para el manejo correcto de archivos y validará situaciones comunes como evitar sobrescritura accidental o errores de apertura.

### Actividades:

1. Crear archivo inicial con productos: Crear un archivo de texto llamado **productos.txt** con tres productos. Cada línea debe tener: **nombre,precio,cantidad**

2. Leer y mostrar productos: Crear un programa que abra **productos.txt**, lea cada línea, la procese con `.strip()` y `.split(",")`, y muestre los productos en el siguiente formato:

Producto: Lapicera | Precio: \$120.5 | Cantidad: 30

3. Agregar productos desde teclado: Modificar el programa para que luego de mostrar los productos, le pida al usuario que ingrese un nuevo producto (nombre, precio, cantidad) y lo agregue al archivo sin borrar el contenido existente.

4. Cargar productos en una lista de diccionarios: Al leer el archivo, cargar los datos en una lista llamada **productos**, donde cada elemento sea un diccionario con claves: **nombre, precio, cantidad**.

5. Buscar producto por nombre: Pedir al usuario que ingrese el nombre de un producto. Recorrer la lista de productos y, si lo encuentra, mostrar todos sus datos. Si no existe, mostrar un mensaje de error.

6. Guardar los productos actualizados: Después de haber leído, buscado o agregado productos, sobrescribir el archivo **productos.txt** escribiendo nuevamente todos los productos actualizados desde la lista.

### Consejo final:

Antes de empezar, analizá cada problema y pensá cómo dividirlo en partes:

- Leer archivo
- Procesar datos
- Mostrar o actualizar información
- Guardar los cambios

Al terminar, probá tu programa varias veces:

- ¿Se puede agregar más de un producto?
- ¿Se guarda todo correctamente?
- ¿Se muestra bien el resultado?