

# ACTIVIDAD 2

## LISTA, DICCIONARIO, OPERADORES, COLLECTIONS

### MARCO ANTONIO VARGAS LIRA

#### 1. Lectura de datos

Se define un diccionario llamado `datos_clientes` que contiene dos listas:

- **compras:** nombres de clientes que hicieron compras (incluye repetidos).
- **registrados:** clientes previamente inscritos en el sistema.

Estas listas se obtienen y se guardan en variables independientes para su análisis.

---

#### 2. Identificación de clientes nuevos

El programa compara la lista de compras con la de registrados:

```
clientes_nuevos = list(set(compras) - set(registrados))
```

Esto permite obtener a los clientes que **compraron pero no están registrados**, es decir, clientes completamente nuevos.

---

### 3. Eliminación de duplicados manteniendo el orden

Para mostrar una lista de compradores sin repetir nombres, se usa:

```
)clientes_unicos = list(OrderedDict.fromkeys(compras))
```

`OrderedDict` conserva el orden original, a diferencia de un set tradicional.

---

### 4. Conteo de compras por cliente

Se utiliza `Counter` para saber cuántas veces aparece cada nombre en la lista de compras:

```
conteo_compras = Counter(compras)
```

Así se identifica cuántas compras hizo cada persona.

---

### 5. Detección de clientes frecuentes

El programa genera un diccionario con los clientes que han comprado **más de una vez**:

```
resumen_frecuentes = {  
    cliente: f"Ha comprado {veces} veces"  
    for cliente, veces in conteo_compras.items()  
    if veces > 1  
}
```

Esto permite reconocer a los clientes con mayor actividad.

---

### 6. Presentación de resultados

Finalmente, imprime tres listas:

1. **Cientes nuevos no registrados.**
  2. **Cientes únicos sin duplicados.**
  3. **Cientes frecuentes (más de una compra).**
- 

## Conclusión

Este código es útil para administrar información de ventas y registro de clientes. Permite identificar nuevos clientes, evitar duplicados, analizar la frecuencia de compras y generar un resumen limpio y ordenado. Es una base funcional para sistemas de ventas, fidelización o CRM.

```
from collections import Counter, OrderedDict

# 1. Diccionario con las listas de compras y registrados
datos_clientes = {
    "compras": [
        "Marco", "Jorge", "Ivonne", "Lucia", "Liliana", "Carlos",
        "Mariana", "Lucia", "Sebas", "Baruc", "Carlos", "Sofia"
    ],
    "registrados": ["Ana", "Pedro", "Maria", "Lucia", "Miguel",
"Carlos"]
}

# Acceso a las listas dentro del diccionario
compras = datos_clientes["compras"]
registrados = datos_clientes["registrados"]

# 2. Filtrar clientes nuevos (no registrados)
clientes_nuevos = list(set(compras) - set(registrados))

# 3. Eliminar duplicados y mantener el orden
clientes_unicos = list(OrderedDict.fromkeys(compras))

# 4. Contar cuántas veces aparece cada cliente
```

```
conteo_compras = Counter(compras)

# 5. Crear resumen personalizado (solo clientes frecuentes)
resumen_frecuentes = {
    cliente: f"Ha comprado {veces} veces"
    for cliente, veces in conteo_compras.items()
    if veces > 1
}

# Formato final de salida
print("=== CLIENTES NUEVOS NO REGISTRADOS ===")
print(clientes_nuevos)

print("\n=== CLIENTES UNICOS (SIN DUPLICADOS) ===")
print(clientes_unicos)

print("\n=== CLIENTES FRECUENTES (MAS DE 1 COMPRA) ===")
for cliente, mensaje in resumen_frecuentes.items():
    print(f"{cliente}: {mensaje}")
```

**<https://github.com/themarco11/Python-INTERMEDIO.git>**