

Práctico 6: Estructuras de datos complejas

Objetivo:

Dominar el uso de estructuras de datos complejas en Python para almacenar, organizar y manipular datos de manera eficiente, aplicando buenas prácticas y optimizando el rendimiento de las aplicaciones.

Resultados de aprendizaje:

1. Comprensión y aplicación de iterables: listas, tuplas y sets.
2. Introducción a estructuras de datos complejas: diccionarios.

Actividades

1) Dado el diccionario precios_frutas

```
precios_frutas = {'Banana': 1200, 'Ananá': 2500, 'Melón': 3000, 'Uva': 1450}
```

Añadir las siguientes frutas con sus respectivos precios:

- Naranja = 1200
- Manzana = 1500
- Pera = 2300

2) Siguiendo con el diccionario precios_frutas que resulta luego de ejecutar el código desarrollado en el punto anterior, actualizar los precios de las siguientes frutas:

- Banana = 1330
- Manzana = 1700
- Melón = 2800

3) Siguiendo con el diccionario precios_frutas que resulta luego de ejecutar el código desarrollado en el punto anterior, crear una lista que contenga únicamente las frutas sin los precios.

4) Escribí un programa que permita almacenar y consultar números telefónicos.

- Permití al usuario cargar 5 contactos con su nombre como clave y número como valor.
- Luego, pedí un nombre y mostrale el número asociado, si existe.

Ejemplo:

```
contactos = {"Juan": "123456", "Ana": "987654"}  
# Consultar: "Juan" → muestra "123456"
```

5) Solicita al usuario una frase e imprime:

- Las palabras **únicas** (usando un set).
- Un diccionario con la cantidad de veces que aparece cada palabra.

Ejemplo:

```
#Entrada -> "hola mundo hola"  
  
#Salida:  
Palabras_únicas: {'hola', 'mundo'}  
Recuento: {'hola': 2, 'mundo': 1}
```

6) Permití ingresar los nombres de 3 alumnos, y para cada uno una **tupla de 3 notas**.
Luego, mostrá el promedio de cada alumno.

Ejemplo:

```
alumnos = {  
    "Sofía": (10, 9, 8),  
    "Luis": (6, 7, 7)  
}
```

7) Dado dos sets de números, representando dos listas de estudiantes que aprobaron Parcial 1 y Parcial 2:

- Mostrá los que aprobaron **ambos parciales**.
- Mostrá los que aprobaron **solo uno de los dos**.
- Mostrá la lista total de estudiantes que aprobaron al menos un parcial (sin repetir).

8) Armá un diccionario donde las claves sean nombres de productos y los valores su stock.
Permití al usuario:

- Consultar el stock de un producto ingresado.
- Agregar unidades al stock si el producto ya existe.
- Agregar un nuevo producto si no existe.

9) Creá una agenda donde las claves sean tuplas de (día, hora) y los valores sean eventos.
Ejemplo:

```
agenda = {  
    ("lunes", "10:00"): "Reunión",  
    ("martes", "15:00"): "Clase de inglés"  
}
```

Permití consultar qué actividad hay en cierto día y hora.

10) Dado un diccionario que mapea nombres de países con sus capitales, construí un nuevo diccionario donde:

- Las **capitales** sean las claves.
- Los **países** sean los valores.

Ejemplo:

```
original = {"Argentina": "Buenos Aires", "Chile": "Santiago"}  
invertido = {"Buenos Aires": "Argentina", "Santiago": "Chile"}
```