

## **Ejercicios: Trabajo con listas en Python**

### **Observación Directa en clase. (Modulo4-OD-01)**

No usar IA de ningún tipo. No usar Internet. Trabajo individual del alumno.

Solo usar el material entregado para resolver los ejercicios. Realizar en clase.

Genera una carpeta por cada ejercicio (introduce el código necesario de ejercicios anteriores, puesto que es un ejercicio incremental).

Para probar tu ejercicio completa cada una de las funciones de cada uno de los ejercicios pedidos en la **plantilla que se te entrega (archivo bibliotecaApp.py)**.

**Entrega un único archivo comprimido con todos los ejercicios y plantilla rellena (bibliotecaApp.py) llamado Mod4\_OD\_01.zip/rar.**

**Nota:** Se entrega el **Reto 8 Biblioteca Digital solucionado** como base de los ejercicios.

Los ejercicios propuestos están basados en una ampliación de dicho reto.

**Ampliando el Reto 8: Biblioteca Digital**, incorporando:

- Diccionarios dentro de diccionarios
- Listas de diccionarios
- Múltiples bibliotecas
- Búsquedas en una o varias bibliotecas

Cada ejercicio tiene **3–4 apartados, puntuación total 2 puntos**

## **GUÍA DE EJERCICIOS DE OBSERVACIÓN DIRECTA EN AULA**

**- Diccionarios en Python -**

**Basados en el Reto 8: Biblioteca Digital (versión ampliada)**

### **Ejercicio 1 – Creación de libros y bibliotecas (2 puntos)**

**Objetivo: practicar diccionarios simples y listas de diccionarios.**

- 1. (0,5 puntos)** Crea una función **crear\_libro(titulo, autor, paginas)** que devuelva un diccionario con los datos del libro.
- 2. (0,5 puntos)** Crea una lista llamada **biblioteca1** con **5 libros** usando la función anterior.
- 3. (0,5 puntos)** Añade un **nuevo campo** opcional a cada libro llamado "**genero**" usando acceso y modificación de diccionarios.
- 4. (0,5 puntos)** Recorre la lista e imprime cada título y su género.

### **Ejercicio 2 – Búsqueda dentro de una biblioteca (2 puntos)**

**Objetivo: trabajar filtrado, funciones y métodos .get().**

- 1. (0,5 puntos)** Crea una función **buscar\_por\_autor(biblio, autor)** que devuelva **una lista** con todos los libros escritos por ese autor.
- 2. (0,5 puntos)** Haz que la función use **.get()** para evitar errores cuando falte algún campo.
- 3. (0,5 puntos)** Prueba la función con **biblioteca1** e imprime los resultados.
- 4. (0,5 puntos)** Si no encuentra ningún libro, devuelve el mensaje "**Autor no encontrado**".

### **Ejercicio 3 – Diccionarios dentro de diccionarios (2 puntos)**

**Objetivo: construir estructuras anidadas más complejas.**

- 1. (0,5 puntos)** Crea un diccionario **biblioteca\_detallada** donde cada clave sea el título del libro y cada valor sea un **diccionario interno** con **autor, páginas y género**.
- 2. (0,5 puntos)** Añade un subdiccionario adicional llamado "**prestamo**" con información como:  
"prestamo": {"disponible": True, "dias\_restantes": 0}
- 3. (0,5 puntos)** Crea una función **info\_libro(diccionario, titulo)** que imprima toda la información del libro incluyendo los datos del préstamo.
- 4. (0,5 puntos)** Si el título no existe, muestra el mensaje "**El libro no está en la biblioteca**".

#### Ejercicio 4 – Múltiples bibliotecas (2 puntos)

Objetivo: manejar una estructura de diccionarios dentro de diccionarios, basada en las técnicas vistas en el documento (anidamiento).

1. (0,5 puntos) Crea **tres bibliotecas** llamadas **biblioteca1**, **biblioteca2** y **biblioteca3**, cada una con al menos 3 libros.
2. (0,5 puntos) Crea un diccionario general llamado **sistema\_bibliotecas** con esta estructura:

```
{  
    "central": biblioteca1,  
    "universitaria": biblioteca2,  
    "infantil": biblioteca3  
}
```
3. (0,5 puntos) Recorre todas las bibliotecas usando **items()** e imprime la cantidad de libros de cada una (usa **len()**).
4. (0,5 puntos) Añade al sistema una **nueva biblioteca** vacía llamada **"digital"**.

#### Ejercicio 5 – Búsqueda global de libros en varias bibliotecas (2 puntos)

Objetivo: combinar diccionarios, listas, funciones, bucles y filtrado.

1. (0,5 puntos) Crea una función **buscar\_libro\_global(sistema, título)** que busque un título en **todas** las bibliotecas del sistema. Debe devolver el nombre de la biblioteca donde se encontró y el diccionario del libro.
2. (0,5 puntos) Crea otra función **buscar\_autor\_global(sistema, autor)** que devuelva **todas** las apariciones del autor en todas las bibliotecas.
3. (0,5 puntos) Permite que el usuario indique **en qué bibliotecas quiere buscar**, enviando una lista de nombres de bibliotecas (por ejemplo: **["central", "infantil"]**).
4. (0,5 puntos) Imprime los resultados formateados así:

Biblioteca: X

Título: Y

Autor: Z

Páginas: W