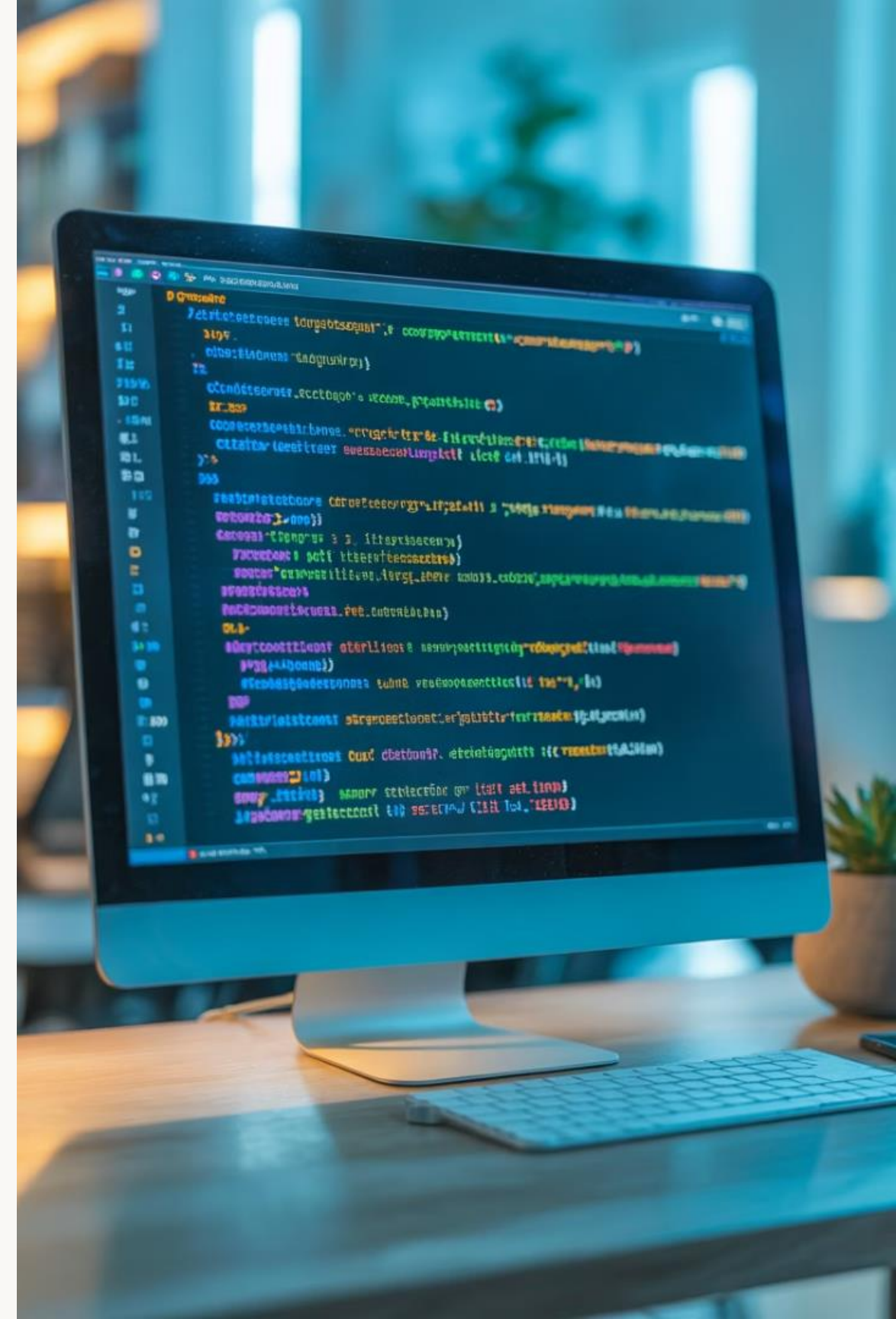


Estructuras de Datos en Python

Exploraremos las principales estructuras de datos en Python. Aprenderemos listas, tuplas, sets y diccionarios con ejemplos prácticos.



A photograph of a wooden desk in a bright room. On the desk, there is a laptop displaying code, several notebooks (one open with 'WEEKLY GOALS' written on it), a pen, and a small potted plant on the windowsill in the background.

Agenda del Curso

1

Repaso de Listas

Características, métodos y ejercicios prácticos

2

Tuplas

Estructuras inmutables y sus aplicaciones

3

Sets

Conjuntos únicos y operaciones matemáticas

4

Diccionarios

Pares clave-valor y ejercicio integrador

¿Qué son las Listas?

Definición

Estructuras que almacenan secuencias ordenadas de elementos. Son muy utilizadas en Python.

Flexibilidad

Permiten elementos de cualquier tipo. Son mutables después de ser creadas.

Acceso

Se accede mediante índices. Se recorren fácilmente con bucles.





Características de las Listas



Ordenadas

Mantienen el orden
de inserción



Mutable

Se pueden modificar
después de crear



Permiten
duplicados

Pueden tener
elementos repetidos



Accesibles por
índice

Uso de números para
acceder

Ejemplo Básico de Lista

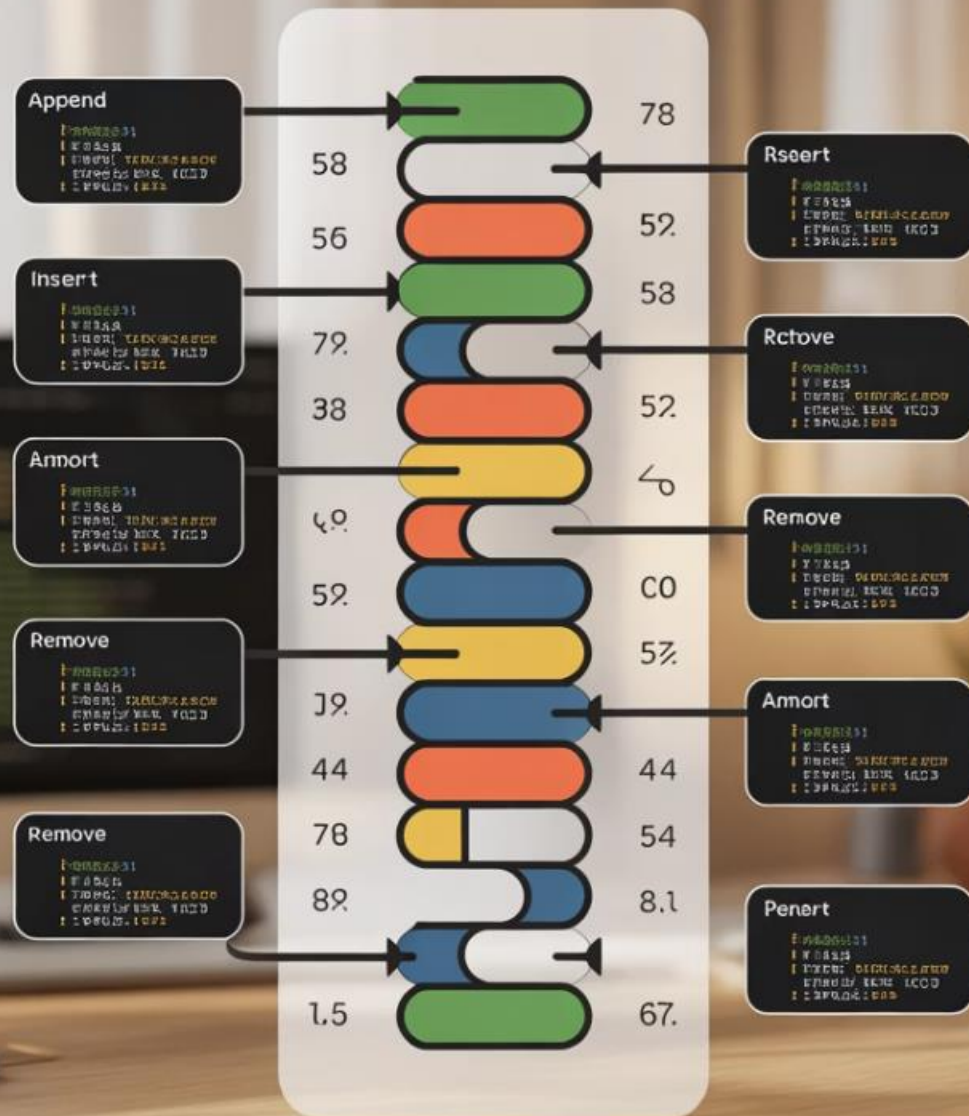
Creación

```
nombres = ["Ana", "Luis", "María"]
```

Modificación

```
nombres.append("Pedro")  
print(nombres[1]) # Imprime "Luis"
```

Python List Methods



Métodos Comunes de Listas

1

`append(valor)`

Agrega elemento al final

2

`insert(pos, valor)`

Inserta en posición específica

3

`remove(valor)`

Elimina un valor específico

4

`pop([pos])`

Elimina y devuelve elemento



Más Métodos de Listas

`sort()`

Ordena la lista alfabéticamente o numéricamente

1

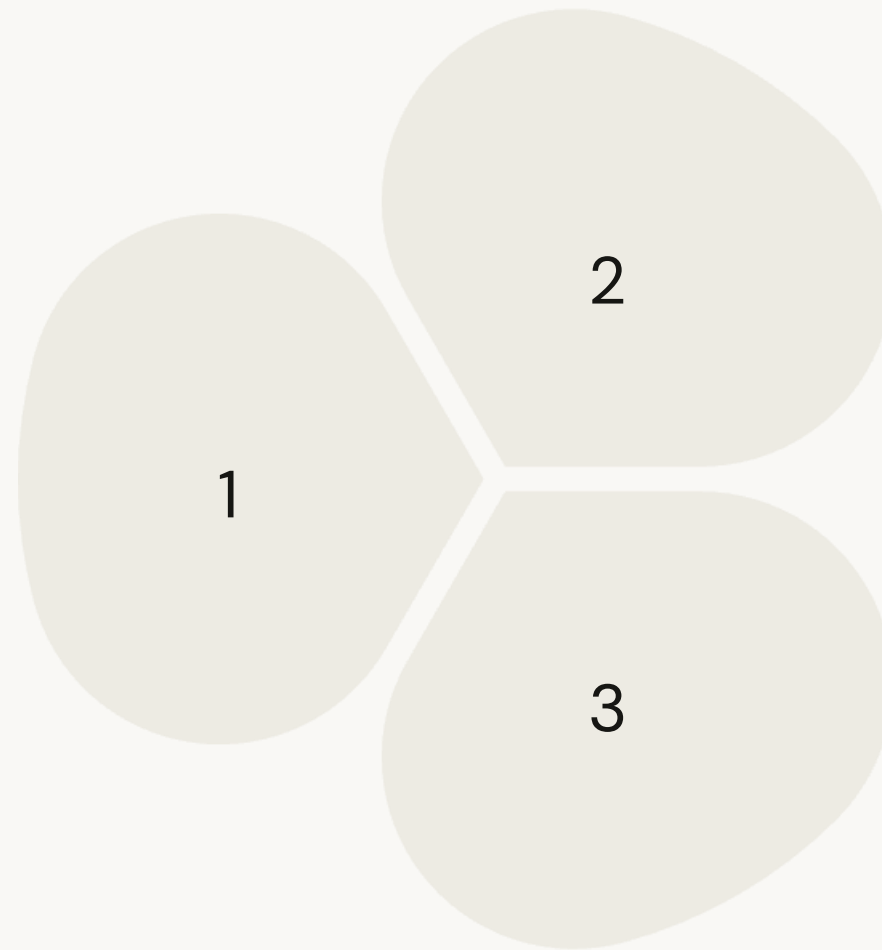
2

`reverse()`

Invierte el orden de los elementos

Uso Típico de Listas

Listas de tareas
Organizar actividades pendientes



Nombres

Almacenar información de personas

Números

Secuencias numéricas y cálculos

Ejercicio 1: Enunciado

Crear lista inicial

5 nombres de estudiantes

Agregar nombre

Nuevo estudiante al final

Eliminar segundo

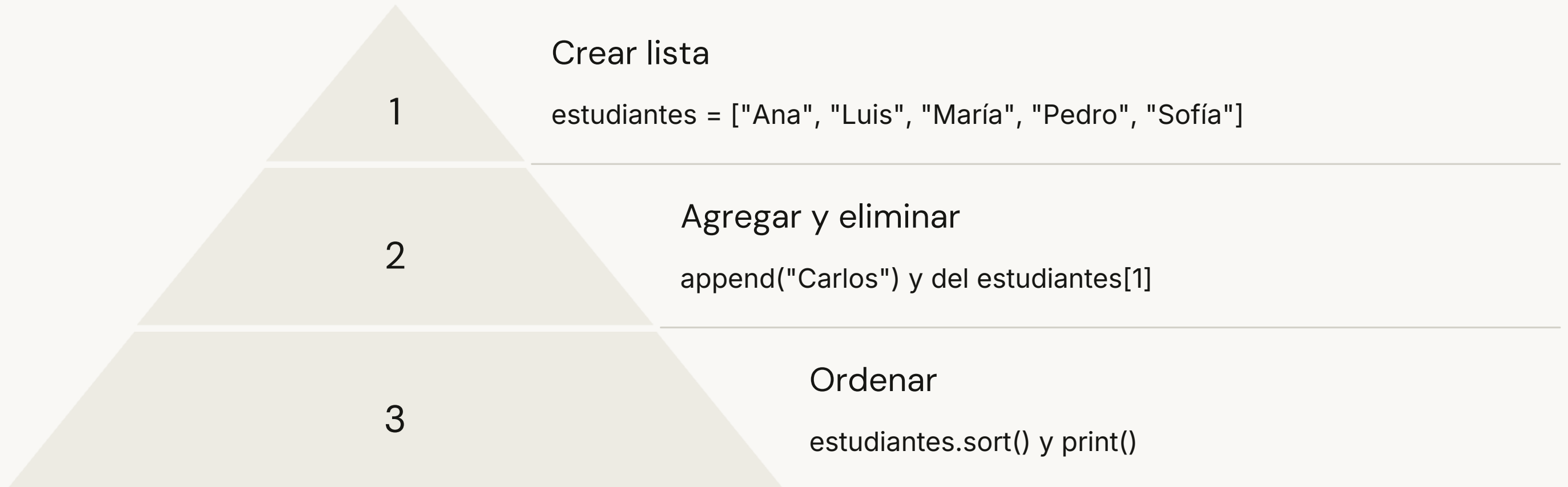
Quitar elemento en posición 1

Ordenar e imprimir

Lista alfabética final



Solución Ejercicio 1



¿Qué son las Tuplas?

1

Inmutables

No se pueden modificar después de crear

2

Ordenadas

Mantienen secuencia de elementos

3

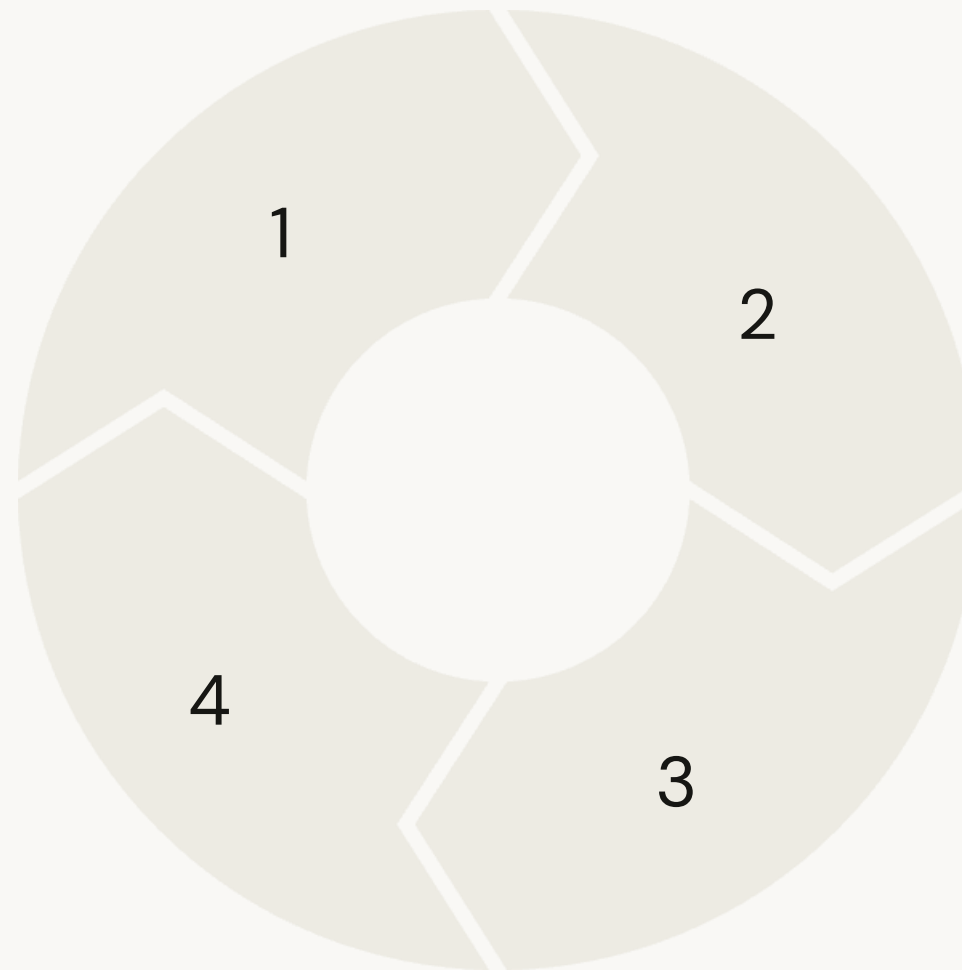
Eficientes

Mayor rendimiento en memoria

Características de Tuplas

Ordenadas
Secuencia fija de elementos

Acceso por índice
Numeración para acceder



Inmutables
No modificables tras creación

Duplicados permitidos
Elementos pueden repetirse

Ejemplo de Tupla

Creación

```
coordenada = (10, 20)  
print(coordenada[0]) # Imprime 10
```

Métodos

count(valor): cuenta apariciones

index(valor): devuelve índice

Ventajas de las Tuplas

2x

Más rápidas

Mayor velocidad de acceso

30%

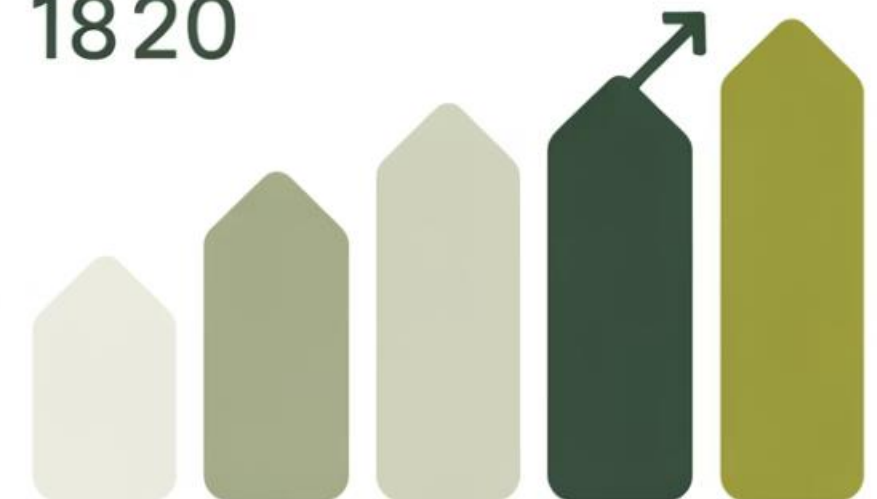
Menos memoria

Menor uso de recursos

Optimized performance

Memory Efficiency

1820



Menery

2.100m

2140k

2300m

100.m



[Request a demo](#)





Ejercicio 2: Catálogo de Productos

Datos

Tupla de tuplas con productos. Nombre, categoría y precio incluidos.

Tareas

Filtrar electrónica, calcular promedio. Encontrar más caro y ubicar Mesa.

Solución: Productos Electrónica

Código

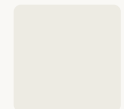
```
for producto in productos:  
    if producto[1] == "Electrónica":  
        print(producto)
```

Resultado

Notebook, Auriculares y Celular. Todos los productos de electrónica mostrados.

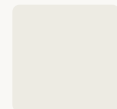


¿Qué son los Sets?



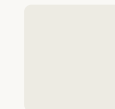
Colección única

No ordenada de elementos
únicos sin duplicados



Operaciones matemáticas

Ideal para unión, intersección y
diferencia



Mutables

Pueden modificarse agregando
o quitando elementos

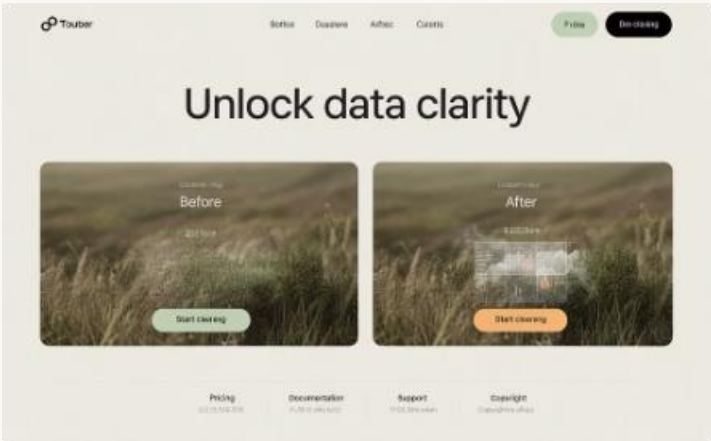
Características de Sets



No ordenados
Sin orden específico de elementos

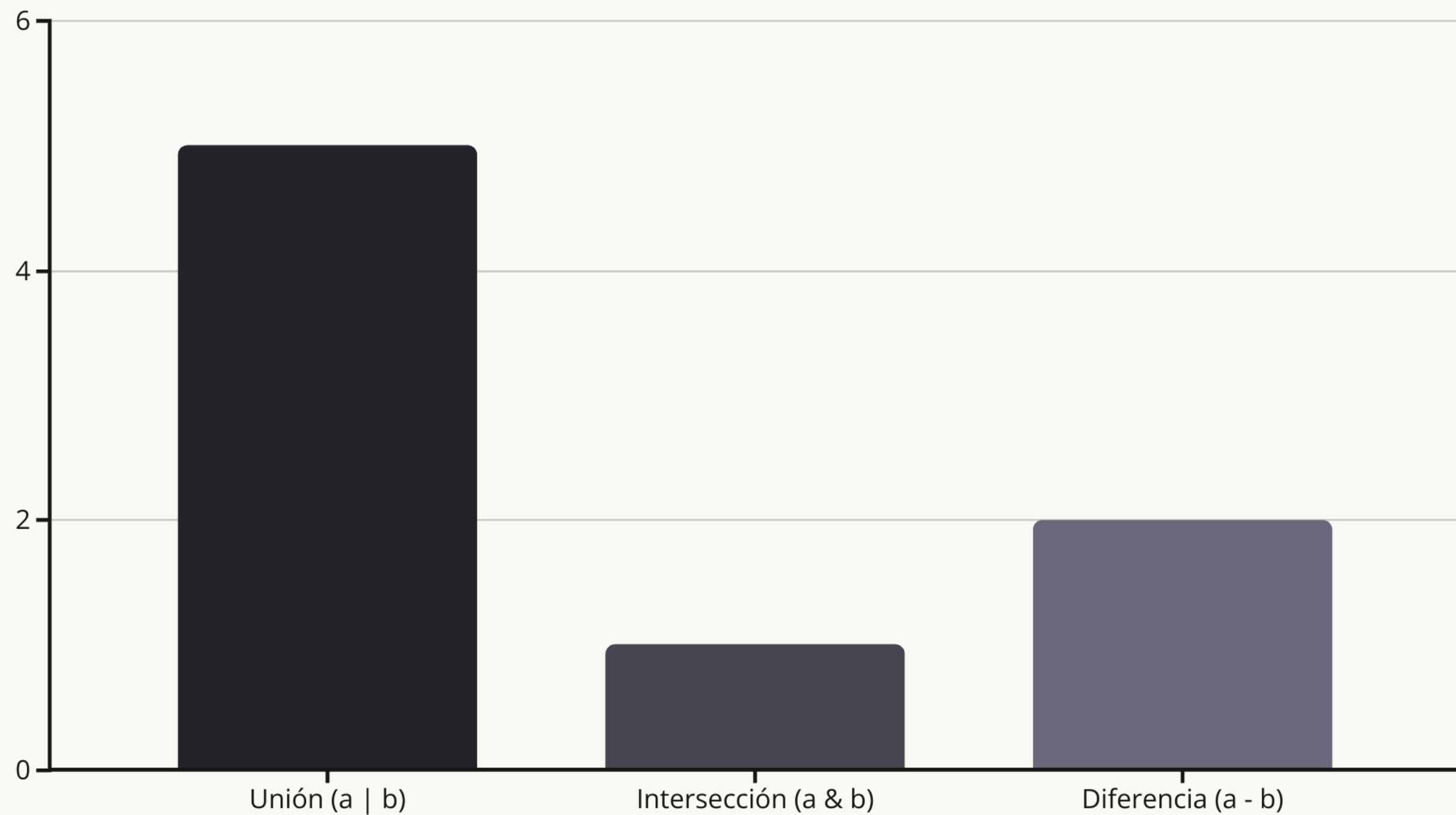



No indexables
No se accede por posición



Sin duplicados
Elementos únicos solamente

Operaciones con Sets





Ejercicio 3: Lenguajes de Programación

1

Crear conjuntos

Estudiante A y B con sus lenguajes

2

Encontrar comunes

Intersección de ambos conjuntos

3

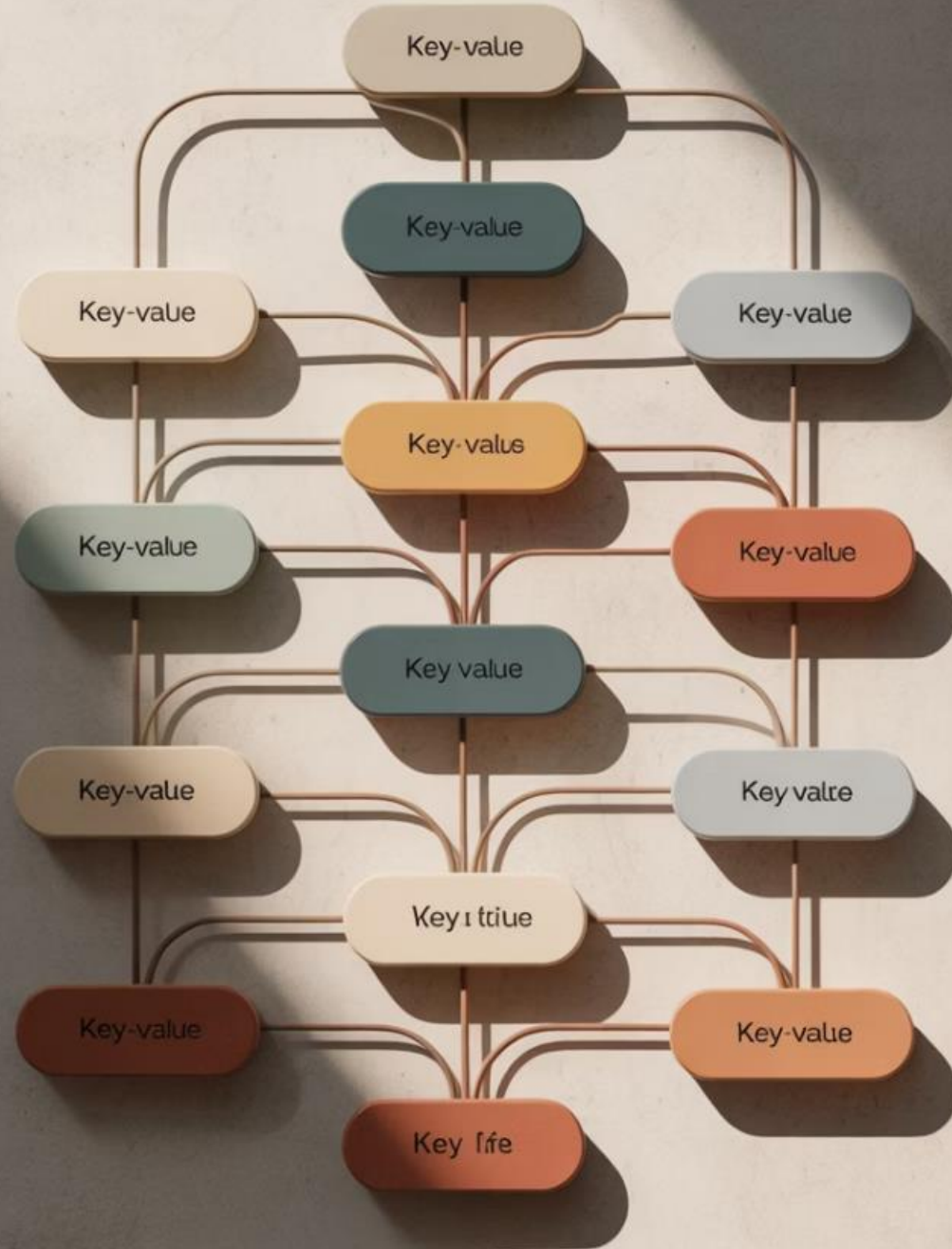
Solo de A

Diferencia entre conjuntos

4

Agregar a B

Nuevo lenguaje al conjunto



¿Qué son los Diccionarios?

1

Pares clave-valor

Estructura de datos asociativa

2

Claves únicas

Identificadores inmutables únicos

3

Valores flexibles

Cualquier tipo de dato

Ejemplo de Diccionario

Creación

```
alumno = {"nombre": "Lucía", "edad": 20,  
"carrera": "Informática"}
```

Acceso

```
print(alumno["nombre"]) # Imprime Lucía
```


Métodos de Diccionarios

Métodos de Acceso

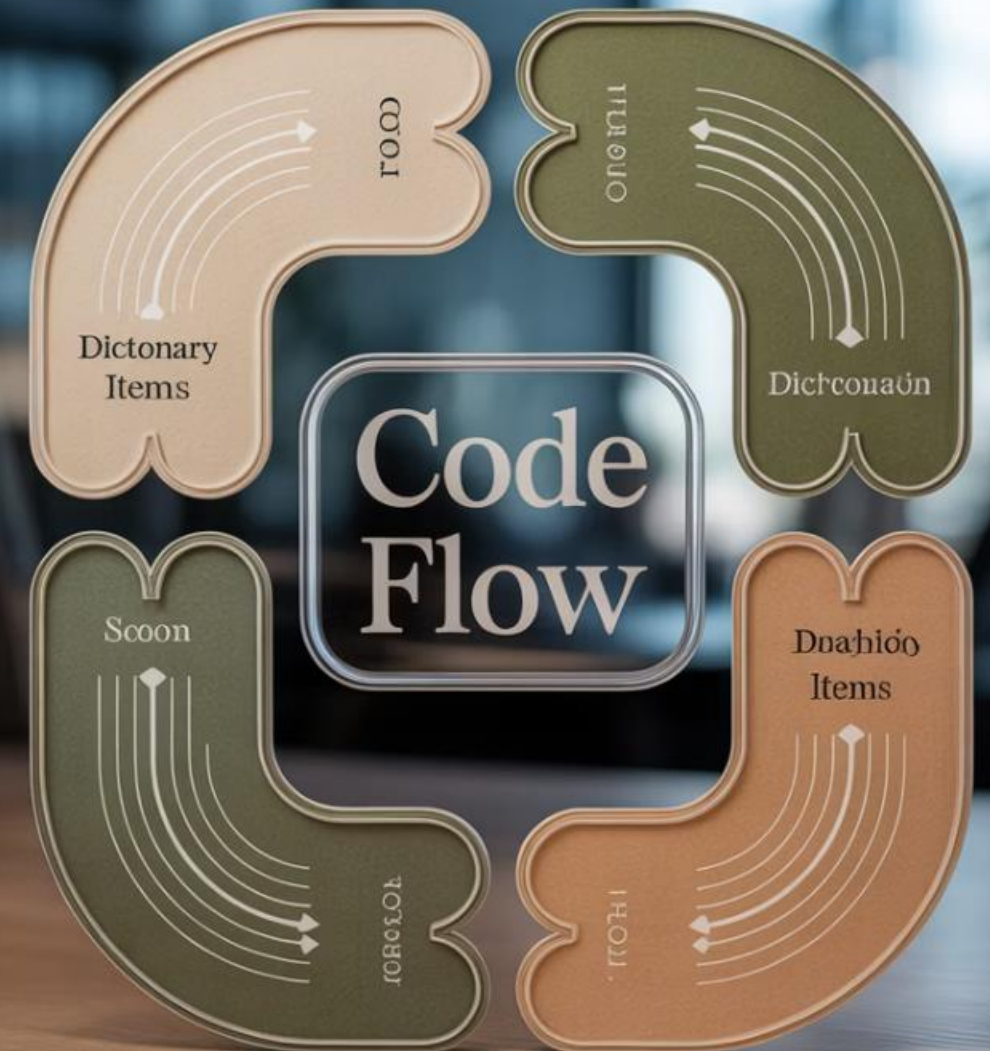
`keys()`, `values()`, `items()` para obtener componentes del diccionario.

Manipulación Segura

`get()` con valor por defecto evita errores de clave inexistente.

Modificación Avanzada

`update()`, `pop()`, `clear()` para gestión completa de datos.



Iteración de Diccionarios

Método items()

```
for clave, valor in alumno.items():  
    print(f"{clave}: {valor}")
```

Ventaja

Acceso simultáneo a claves y valores. Forma más eficiente de recorrer.

```
alumno = {  
    "nombre": "Lucía",  
    "edad": 20,  
    "carrera": "Informática"  
}
```

```
#Recorrer por clave  
for clave in alumno.keys():  
    print(clave)
```

```
#Recorrer por valor  
for valor in alumno.values():  
    print(valor)
```

```
#Recorrer por clave y valor  
for clave, valor in alumno.items():  
    print(clave, valor)
```

Ejercicio 4: Diccionario de Libro

Crear diccionario

Libro con título, autor, año, género

Mostrar datos

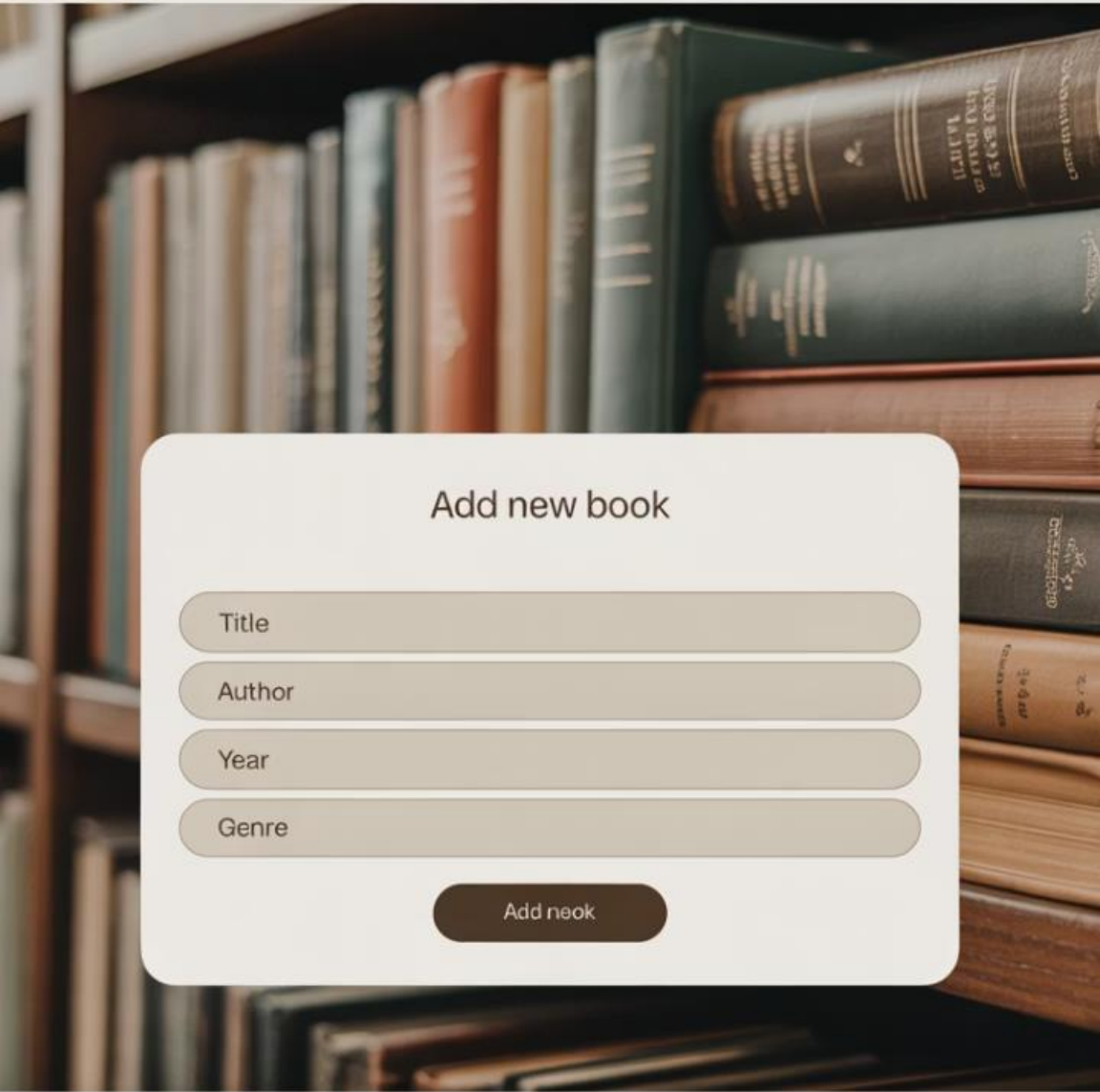
Recorrer con bucle items()

Agregar disponible

Nueva clave con valor True

Cambiar año

Modificar valor existente



Add new book

Title

Author

Year

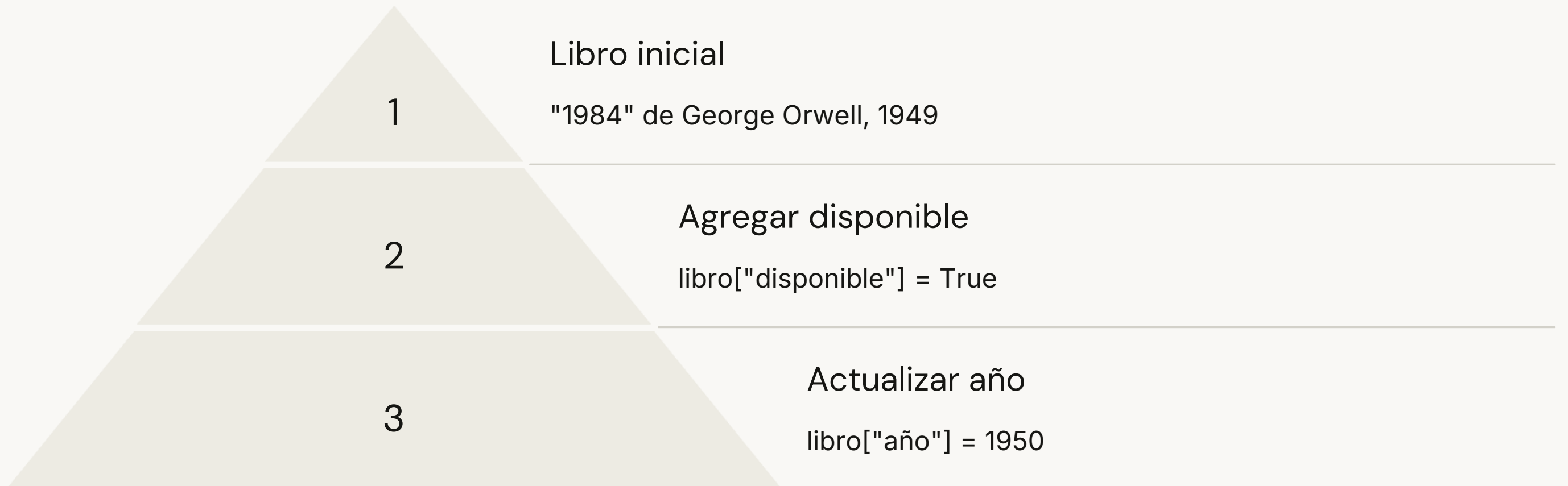
Genre

Add neok

edlyr Book

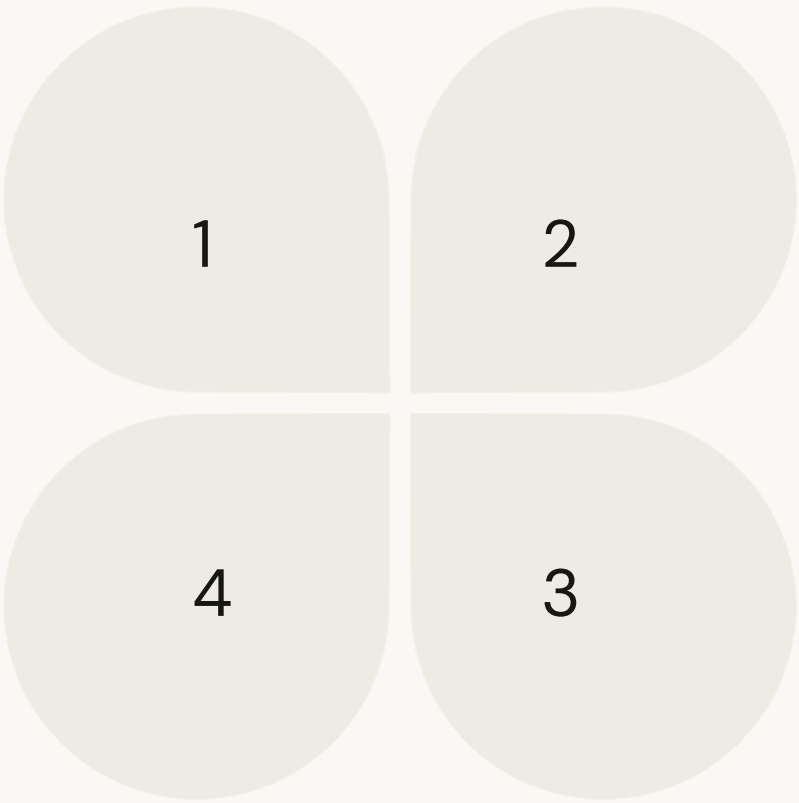
Title	Author	Year	🔒	⌵
Author	Author	Genre	🔒	⌵
Yile	Year	Genre	🔒	⌵
Pedl	Year	Genre	🔒	⌵
	Yntho	Geneco		

Solución Ejercicio 4



Ejercicio Integrador: Escuela

Estudiantes únicos
Usar set para eliminar duplicados

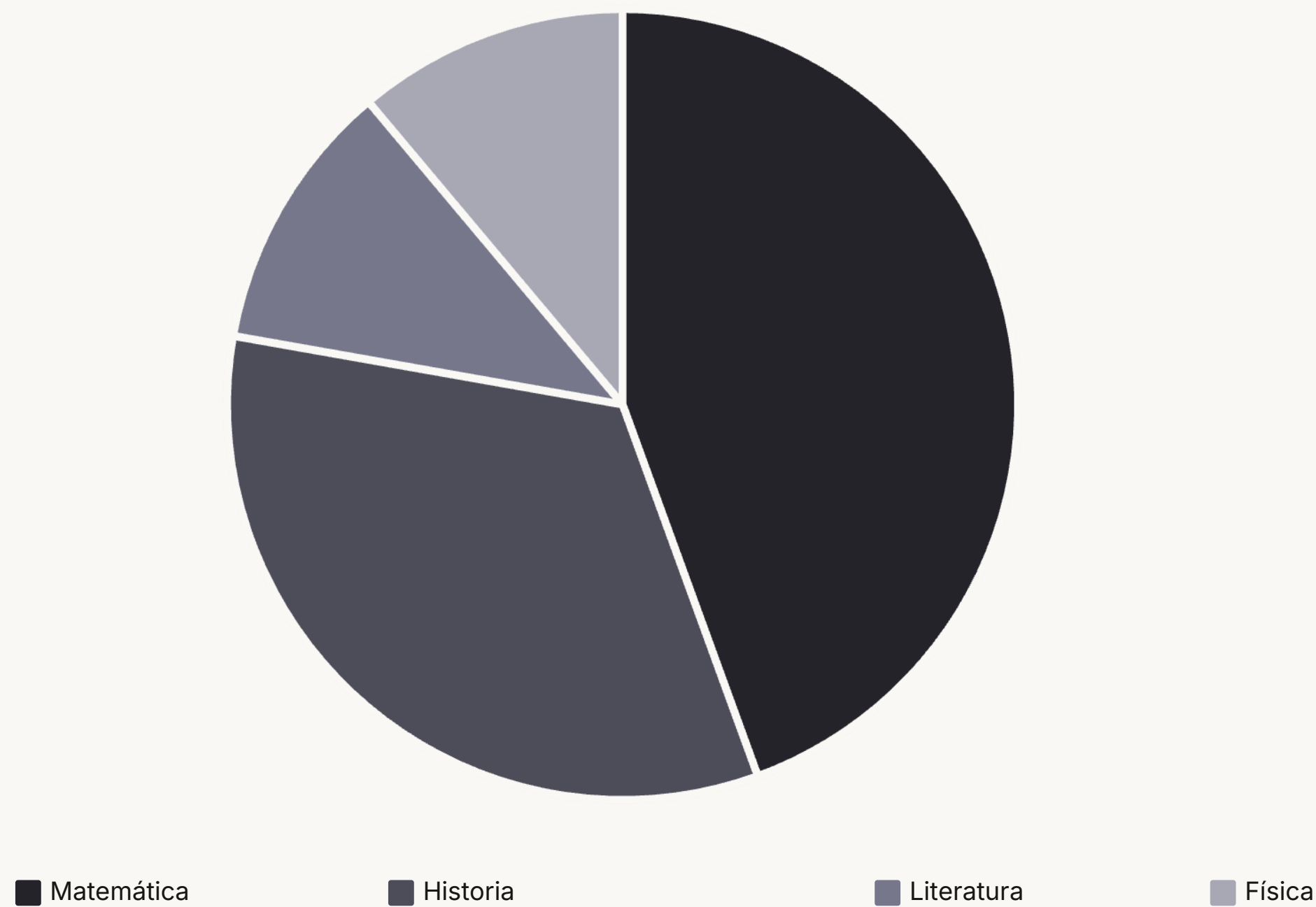


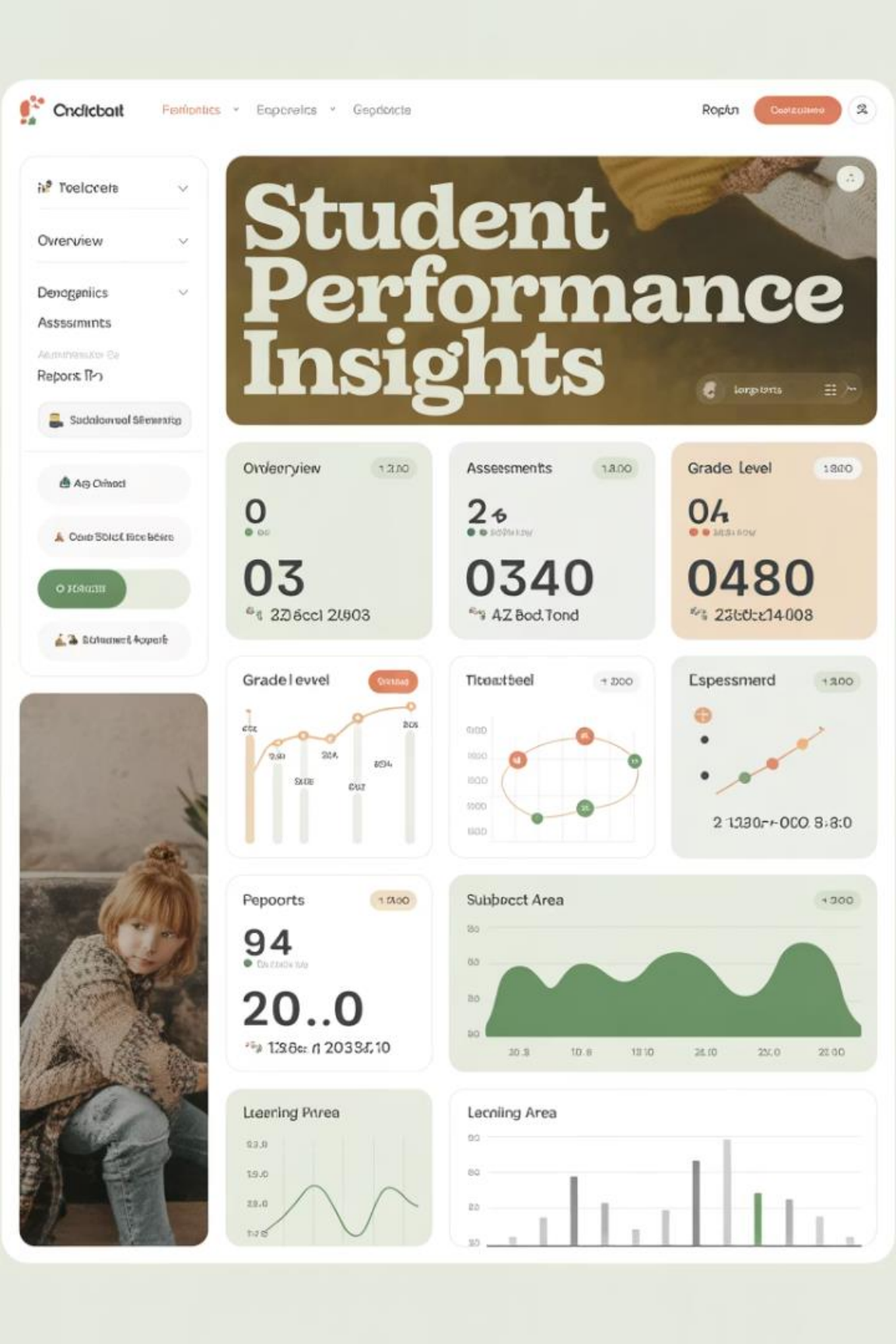
Materias por estudiante
Diccionario con conteos

Materias únicas
Exclusivas de cada estudiante

Materias comunes
Intersección de conjuntos

Datos del Ejercicio Integrador





Resultados del Ejercicio

4

Estudiantes

Ana, Luis, Carlos, Sofía

4

Materias

Matemática, Historia, Literatura, Física

0

Comunes

Ninguna materia cursada por todos

2

Únicas

Literatura y Física exclusivas



Resumen y Próximos Pasos

- | | |
|---|---|
| <div data-bbox="166 996 274 1100">■</div> <div data-bbox="320 1020 954 1072">Estructuras dominadas</div> <div data-bbox="320 1131 1349 1253">Listas, tuplas, sets y diccionarios completados exitosamente</div> | <div data-bbox="1567 996 1675 1100">■</div> <div data-bbox="1721 1020 2254 1072">Ejercicios resueltos</div> <div data-bbox="1721 1131 2772 1183">Práctica con casos reales y ejercicio integrador</div> |
| <div data-bbox="166 1364 274 1468">■</div> <div data-bbox="320 1388 711 1440">Siguiendo nivel</div> <div data-bbox="320 1499 1463 1551">Aplicar conocimientos en proyectos más complejos</div> | <div data-bbox="1567 1364 1675 1468">■</div> <div data-bbox="1721 1388 2356 1440">Continuar aprendiendo</div> <div data-bbox="1721 1499 2707 1551">Explorar algoritmos y estructuras avanzadas</div> |