

EJERCICIO PREPARACIÓN

EJERCICIO

Los clubes de aventuras suelen organizar sesiones donde sus integrantes participan de diferentes desafíos. Cada aventurero se identifica por un código único, posee un nombre y una edad, y va acumulando puntajes a lo largo de distintas sesiones.

Este ejercicio consiste en diseñar un sistema que permita registrar aventureros, almacenar sus puntajes por sesión, modificar estos puntajes y consultar estadísticas de participación. El objetivo es aplicar conocimientos de estructuras de datos, funciones y validación de entradas.

Se desea implementar un sistema de gestión para un **Club de Aventureros** que permita, a través de un menú de opciones cumplir con la siguiente funcionalidad:

1. Registrar aventureros con su código único, nombre y edad.
2. Registrar puntajes obtenidos por sesión para cada aventurero.
3. Modificar un puntaje específico de una sesión dada.
4. Visualizar el total acumulado y el promedio de puntajes por aventurero.
5. Mostrar los aventureros cuyo promedio de puntajes esté por debajo de un umbral dado.
6. Listar todos los aventureros con sus datos y puntajes.
7. Filtrar los aventureros por edad específica.

El menú deberá mantenerse activo hasta que el usuario decida salir.

ESTRUCTURA DE LA COLECCIÓN

aventureros = {

"AV001": {

"nombre": "Luna",

"edad": 12,

"puntajes": [75, 82, 91]

```
},  
  
...  
  
}
```

REQUERIMIENTOS DE CONSTRUCCIÓN

Las funciones mínimas que se deben implementar son:

- registrar_aventurero(lst_aventureros: dict, codigo: str, datos: list) -> bool
- registrar_puntaje(lst_aventureros: dict, codigo: str, puntaje: int) -> bool
- modificar_puntaje(lst_aventureros: dict, codigo: str, sesion: int, nuevo_puntaje: int) -> bool
- mostrar_participacion(lst_aventureros: dict)
- participantes_con_bajo_promedio(lst_aventureros: dict, umbral: float)
- listar_aventureros(lst_aventureros: dict)
- obtener_aventureros_por_edad(lst_aventureros: dict, edad: int)

RESTRICCIONES

- El código del aventurero debe ser único.
- No se pueden registrar puntajes si el aventurero no existe.
- No se puede modificar un puntaje si la sesión no existe o el código es incorrecto.
- En las funciones que devuelven True o False, el valor debe reflejar el éxito o fracaso de la operación.
- Se debe usar manejo de errores (como try/except) en la interacción con el usuario.
- Los puntajes deben ser numéricos y mayores o iguales a cero.