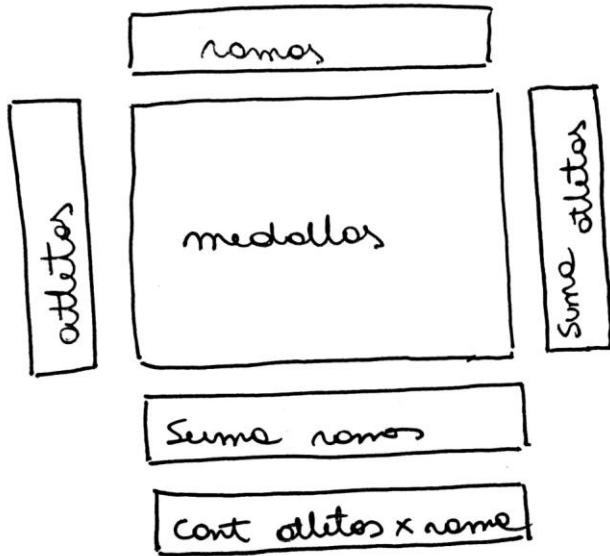


Desafío 9 (posible solución)

Parte a. Estructura de datos



EXPLICACIÓN DE LA ESTRUCTURA

Ramas: esta lista guarda el nombre de todas las ramas

Atletas: esta lista guarda el nombre de todos los atletas

Medallas: corresponde a la matriz que guardará la cantidad de medallas ganadas por atleta en cada rama

Suma Atletas: contiene la suma de todas las medallas ganadas por cada atleta

Suma Ramas: contiene la suma de todas las medallas obtenidas por cada rama

Cantidad Atletas por Rama: guarda la cantidad de atletas participantes en cada rama

Parte b. Código Python

```
1. import numpy
2. listaAtletas = []
3. listaRamas = []
4. matriz = numpy.zeros([10,10])
5.
6. atleta = input("ingrese nombre atleta, fin para terminar")
7.
8. while atleta != "fin":
9.
10.     if not atleta in listaAtletas:
11.         listaAtletas.append(atleta)
12.
13.     rama = input("ingrese nombre rama, fin para terminar")
14.
15.     while rama != "fin":
16.         cantMedallas = int(input("ingrese cantidad de medallas"))
17.
18.         if not rama in listaRamas:
19.             listaRamas.append(rama)
20.
21.         fila = listaAtletas.index(atleta)
22.         columna = listaRamas.index(rama)
23.         matriz[fila][columna] += cantMedallas
24.
25.         rama = input("ingrese nombre rama, fin para terminar")
26.
27.     atleta = input("ingrese nombre atleta, fin para terminar")
28.
29. #INICIO REQUERIMIENTO A
30. print("Se despliegan las ramas y las sumas de medallas")
31. listaSumaRamas = []
32. listaCantAtletas = []
33. for x in range(len(listaRamas)):
34.     suma = 0
35.     cantAtletas = 0
36.     for y in range(len(listaAtletas)):
37.         suma = suma + matriz[y][x]
38.         if matriz[y][x] != 0:
```

```

39.         cantAtletas += 1
40.     listaSumaRamas.append(suma)
41.     listaCantAtletas.append(cantAtletas)
42.
43. for x in range(len(listaRamas)):
44.     print(listaRamas[x]+" con "+str(listaSumaRamas[x]))
45. #FIN REQUERIMIENTO A
46.
47. #INICIO REQUERIMIENTO B
48. listaSumaAtletas = []
49. for x in range(len(listaAtletas)):
50.     suma = 0
51.     for y in range(len(listaRamas)):
52.         suma = suma + matriz[x][y]
53.     listaSumaAtletas.append(suma)
54.
55. #en la variable atletaMayor buscamos el valor mayor en la lista
56. #que contiene la suma de medallas por atleta.
57. atletaMayor = max(listaSumaAtletas)
58. #Hacemos un ciclo para comprobar si es que hay más de un atleta
59. #con la cantidad de medallas mayor
60. for x in range(len(listaAtletas)):
61.     if listaSumaAtletas[x] == atletaMayor:
62.         print("Atleta que tuvo la cantidad mayor de medallas")
63.         print(listaAtletas[x])
64.         #buscamos la rama con la menor cantidad de medallas para el
65.         #atleta mayor
66.         cantMenor = min(matriz[x])
67.         print("el menor es" + str(cantMenor))
68.         #hacemos un ciclo para comprobar si es que hay más de una rama
69.         #con la cantidad de medallas menor
70.         print("sus ramas menores fueron")
71.         for y in range(len(listaRamas)):
72.             if matriz[x][y] == cantMenor:
73.                 print(">" + listaRamas[y])
74. #FIN REQUERIMIENTO B
75.
76. #INICIO REQUERIMIENTO C
77. #Este requerimiento es igual que el requerimiento B con la diferencia
78. #de que usamos las listas paralelas de las ramas
79. ramaMenor = min(listaCantAtletas)
80. print("Rama o ramas olímpicas con la cantidad menor de participantes")
81. for x in range(len(listaRamas)):
82.     if listaCantAtletas[x] == ramaMenor:
83.         print(listaRamas[x])
84.
85.         #debemos buscar la cantidad mayor en la columna de la rama
86.         cantMayor = -1
87.         for y in range(len(listaAtletas)):
88.             if matriz[y][x] > cantMayor:
89.                 cantMayor = matriz[y][x]
90.
91.         #ahora que hemos encontrado el atleta mayor, debemos buscar
92.         #si hay más de un atleta con dicho valor
93.         for z in range(len(listaAtletas)):
94.             if matriz[z][x] == cantMayor:
95.                 print(">", listaAtletas[z])
96. #FIN REQUERIMIENTO C

```