

## Programación – Certamen 2 - Jueves 27 de Junio de 2024

Se está desarrollando un estudio sobre el tiempo de uso del celular por parte de la población de Pythonia. Para ello, se ha recolectado información sobre las sesiones de uso del móvil. Una sesión corresponde a cada ocasión en que una persona utiliza su teléfono con alguna aplicación. La información se encuentra almacenada en un archivo que contiene una línea por cada sesión, sin ningún orden particular, siguiendo el siguiente formato:

```
id_usuario;fechahora;duración;app
```

El campo `id` contiene el identificador de la persona, mientras que `fechahora` almacena la fecha y la hora de inicio de la sesión y `duración` contiene la cantidad de minutos que duró. Este valor puede contener decimales. Finalmente, el campo `app` indica la aplicación que se usó durante la sesión.

A continuación se muestra un extracto del archivo descrito:

`uso_celular.txt`

```
18;2024-01-01 13:12;3.89;Candy Crush
2;2024-08-14 21:54;8.43;Brawl Stars
6;2024-04-22 22:21;10.86;WhatsApp
18;2024-05-07 01:42;1.21;Spotify
18;2024-12-31 23:58;10.9;ChatGPT
4;2024-03-08 06:58;0.11;Facebook
17;2024-10-28 20:36;1.91;WhatsApp
17;2024-08-21 04:50;1.05;TikTok
17;2024-08-11 00:50;0.55;YouTube
...
```

Un segundo archivo contiene información sobre las personas, vinculando los identificadores con sus nombres y con la compañía de telefonía que les presta el servicio. Cada línea corresponde a una persona, siguiendo el formato:

```
id_usuario;nombre;proveedor
```

A continuación se muestra un extracto de este archivo:

`usuarios.txt`

```
1;Alejandro;Movistar
2;Juan;Claro
3;Isabel;WOM
4;Sofía;Entel
5;Marta;Movistar
...
```

**Nota:** Puede suponer que toda persona que aparece en el primer archivo también aparece en el segundo.

1. [40%] Escriba la función `ranking_de_uso(na_uso, na_usuarios, mes, año)`, que recibe como parámetro los nombres de los archivos que contienen los datos del uso del celular y de los usuarios, siguiendo en ambos casos el formato descrito anteriormente, y dos números enteros (*int*) que especifican un mes y un año particulares.

La función debe crear y retornar una lista de listas, en la que cada sublista corresponde a un usuario que utilizó su celular en el mes y año indicados. La lista debe estar ordenada de mayor a menor, según el tiempo total de uso del celular. Para cada persona **se debe incluir el nombre y no el identificador**.

Ejemplo:

```
>>> print(ranking_de_uso("uso_celular.txt", "usuarios.txt", 6, 2024))
[[59.06, 'Alejandro'], [44.63, 'Isabel'], [41.81, 'Laura'], [39.55, 'Daniel'],
[29.61, 'Paula'], [27.27, 'Francisco'], [26.27, 'José'], [20.8, 'Manuel'],
[20.27, 'Luis'], [18.92, 'Ana'], [17.89, 'Carmen'], [15.39, 'Javier'],
[14.18, 'David'], [7.61, 'Carlos'], [6.88, 'Lucía'], [5.12, 'María'], [4.62, 'Sofía'],
[2.93, 'Juan'], [1.37, 'Elena']]
```

2. [60 %] Escriba la función `sesiones_por_app(nombre_archivo)`, que recibe como parámetro el nombre del archivo que contiene los datos de uso del celular, siguiendo el formato descrito anteriormente.

La función debe crear un archivo para cada aplicación distinta que se usó, con los datos de las 5 sesiones más largas de la aplicación, ordenadas de mayor a menor según la duración, e identificadas por un correlativo al inicio de cada línea.

El nombre de cada archivo debe ser igual al nombre de la aplicación seguido del sufijo `.txt`. Por ejemplo: `"TikTok.txt"`.

Las líneas de los archivos deben seguir el formato mostrado en los ejemplos, el que incluye la fecha y hora de la sesión, la duración y el identificador del usuario.

Además, la función debe retornar la cantidad de archivos que se crearon.

Ejemplo:

```
>>> print(sesiones_por_app("uso_celular.txt"))  
22
```

A continuación, a modo de ejemplo y para mostrar el formato que se debe seguir, se muestran dos de los archivos que se deberían haber creado:

Candy Crush.txt

```
1. Fecha y hora: 2024-02-01 09:45, duración: 24.81, usuario: 1.  
2. Fecha y hora: 2024-06-18 20:40, duración: 20.15, usuario: 3.  
3. Fecha y hora: 2024-10-13 00:49, duración: 17.48, usuario: 15.  
4. Fecha y hora: 2024-05-06 07:49, duración: 12.6, usuario: 14.  
5. Fecha y hora: 2024-12-14 09:34, duración: 11.41, usuario: 17.
```

Instagram.txt

```
1. Fecha y hora: 2024-09-15 21:02, duración: 11.15, usuario: 19.  
2. Fecha y hora: 2024-08-02 10:15, duración: 10.46, usuario: 10.  
3. Fecha y hora: 2024-05-02 18:45, duración: 9.41, usuario: 13.  
4. Fecha y hora: 2024-10-13 17:18, duración: 8.84, usuario: 13.  
5. Fecha y hora: 2024-11-09 01:41, duración: 8.8, usuario: 3.
```

**Nota:** Es importante recordar que para que el ordenamiento de valores funcione adecuadamente debe trabajarse con tipos numéricos (*int* o *float*) y no con *strings*. Esto es aplicable a ambas preguntas.

**Nota:** No se preocupe por redondear los resultados de las operaciones aritméticas. Para efectos de esta evaluación no es necesario.