Está prohibido usar el módulo Pandas. De hacerlo, la nota final será 0.

Sí pueden utilizar el módulo statistics.

El código debe estar debidamente comentado explicando lo que hace en cada parte de su solución. De no hacerlo, no se considerará el desarrollo del ítem.

Por otro lado, es obligatorio presentar un informe con capturas de sus resultados parciales y finales. **De no hacerlo, la nota final será 0.**

Contexto:

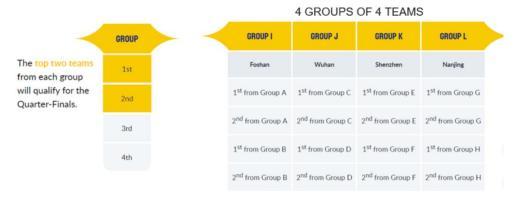
- En el presente laboratorio, se le pide realizar la simulación de una competencia deportiva.
- Para la realización, debe considerar que el torneo cuenta con 32 participantes (equipos) y tendrá tanto fase de grupos como etapa de eliminación directa.

Nota: El contexto de este laboratorio es ficticio.

Fase Grupos

Grupo A	Grupo B	Grupo C	Grupo D
ATL	DAL	LAC	NOP
BOS	DEN	LAL	NYK
BRK	DET	MEM	ОКС
CHA	GSW	MIA	ORL
Grupo E	Grupo F	Grupo G	Grupo H
<mark>Grupo E</mark> POR	Grupo F WAS	Grupo G HOU	Grupo H PHI
POR	•		
•	WAS	HOU	PHI

Segunda ronda



Fase Final



Se le pide elaborar 2 archivos: **servidor_códigoPUCP.py** y **cliente_códigoPUCP.py**. Estos deben cumplir los siguientes requerimientos:

- Los partidos durarán 150 ms (sleep de 0.15 s)
- En la fase de grupos, el ganador del partido es quien tenga mayor puntaje promedio. **Hint: usar la columna "PTS"**
- En la fase de eliminatorias, el ganador del partido se define de manera aleatoria

La interacción del cliente se realizará desde terminal:

servidor_códigoPUCP.py	cliente_códigoPUCP.py
a) Leer el archivo 'datos.csv' (1 pts)	h) Enviar su nombre al servidor y
	esperar el mensaje de confirmación:
	"Procesando data". (0.5pts)

Para cada equipo, generar una lista de 5 jugadores que pertenezcan al mismo equipo y tengan el valor más alto de minutos jugados "MIN". Las listas deben ser globales. (2 pts)	i) Enviar el mensaje "equipos" al servidor para que el servidor genere las listas de b). (0.5pts)
Desarrollar la función partido(equipo1,equipo2): Debe elegir el ganador de forma aleatoria y retornar el nombre del ganador (2 pts)	j) Enviar el mensaje "fase de grupos asincrono" al servidor para que ejecute el apartado d) e imprimir en el cliente la lista clasificados_async. (0.5pts)
Desarrollar la función grupos_async() que retornará la lista clasificados_async con los 2 primeros lugares de cada grupo. Nota: Deberá asumir que los 6 partidos de cada grupo se juegan en simultáneo. (2.5 pts)	k) Enviar el mensaje "fase de grupos sincrono" al servidor para que ejecute el apartado e) e imprimir en el cliente la lista clasificados_sync. (0.5 pts)
Desarrollar la función grupos_sync() que retornará la lista clasificados_sync con los 2 primeros lugares de cada grupo. Nota: Deberá asumir que no hay partidos en simultáneo. Como son 6 partidos por cada grupo y son 8 grupos, los 48 partidos serán uno luego de otro. (2.5 pts)	I) Enviar el mensaje "eliminatorias asincrono" al servidor para que ejecute el apartado f) e imprimir en el cliente la lista podio_async. (0.5 pts)
Desarrollar la función eliminatorias_async() que calculará los resultados de la segunda ronda, cuartos de final, semifinales, tercer lugar y campeón. Retornará la lista: podio_async que tendrá la forma [1° lugar, 2° lugar, 3° lugar]. Nota: En este ítem los partidos de cada etapa se juegan en simultáneo y una etapa debe terminar antes de que comience la	m) Enviar el mensaje "eliminatorias sincrono" al servidor para que ejecute el apartado g) e imprimir en el cliente la lista podio_sync. (0.5 pts)
Desarrollar la función eliminatorias_sync() que calculará los resultados de los octavos de final, cuartos de final, semifinales, tercer lugar y campeón. Retornará la lista: podio_sync que tendrá la forma [1° lugar, 2° lugar, 3° lugar]. (2 pts) Nota: Deberá asumir que no hay partidos en simultáneo y una etapa debe terminar	n) Enviar el mensaje "reporte" al servidor para que el servidor genere el archivo "reporte.txt" que realice una comparación de las listas obtenidas en los ítems d), e), f) y g) y los tiempos de ejecución. (1.5 pts)
	y tengan el valor más alto de minutos jugados "MIN". Las listas deben ser globales. (2 pts) Desarrollar la función partido(equipo1,equipo2): Debe elegir el ganador de forma aleatoria y retornar el nombre del ganador. (2 pts) Desarrollar la función grupos_async() que retornará la lista clasificados_async con los 2 primeros lugares de cada grupo. Nota: Deberá asumir que los 6 partidos de cada grupo se juegan en simultáneo. (2.5 pts) Desarrollar la función grupos_sync() que retornará la lista clasificados_sync con los 2 primeros lugares de cada grupo. Nota: Deberá asumir que no hay partidos en simultáneo. Como son 6 partidos por cada grupo y son 8 grupos, los 48 partidos serán uno luego de otro. (2.5 pts) Desarrollar la función eliminatorias_async() que calculará los resultados de la segunda ronda, cuartos de final, semifinales, tercer lugar y campeón. Retornará la lista: podio_async que tendrá la forma [1° lugar, 2° lugar, 3° lugar]. Nota: En este ítem los partidos de cada etapa se juegan en simultáneo y una etapa debe terminar antes de que comience la siguiente (2 pts) Desarrollar la función eliminatorias_sync() que calculará los resultados de los octavos de final, cuartos de final, semifinales, tercer lugar y campeón. Retornará la lista: podio_sync que tendrá la forma [1° lugar, 2° lugar, 3° lugar]. (2 pts) Nota: Deberá asumir que no hay partidos

Finalmente, debe responder lo siguiente:

¿Qué diferencias puede notar entre la implementación síncrona y asíncrona? ¿A qué se debe la diferencia entre los tiempos de ejecución? (1.5 pts)