

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA**

ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS

**Laboratorio 5
(Semestre 2023-0)**

Indicaciones generales:

- Debe enviar su solución, siguiendo el formato indicado en cada pregunta.
 - De no seguir estas indicaciones tendrá una penalidad en su puntaje.
 - **La hora de entrega es hasta culminada las 2 horas de evaluación de laboratorio.** Posterior a dicha hora, se anulará la evaluación.
 - Cualquier indicio de plagio resultará en la anulación de la prueba.
 - **La presentación, la ortografía y la gramática (incluyendo comentarios) influyen en la calificación.**
-

Experiencia 1: Uso de ayncio

Se le pide realizar la simulación del campeonato mundial de ajedrez que se realiza en 2 etapas: fase de rondas y fase final

Los 4 participantes en la fase de rondas son los siguientes:

-Magnus

-Vladimir

-Peter

-Levon

A continuación, se describen las 2 etapas:

Fase de rondas

Son 3 rondas y en cada ronda se juegan 2 partidas. Como en total son $3 \times 2 = 6$ partidas, al final de las 3 rondas todos habrán jugado contra todos. Cada ronda se juega en días distintos.

Ronda 1 – Día 1

Vladimir	vs	Magnus
Peter	vs	Levon

Ronda 2 – Día 2

Magnus	vs	Peter
Levon	vs	Vladimir

Ronda 3 – Día 3

Magnus	vs	Levon
Peter	vs	Vladimir

Indicaciones para esta fase de rondas:

-Para simular las partidas, cada una durará 0.15 segundos, para ello usará la función sleep de Python.

- Por cada partida ganada el ganador recibe 1 punto, y el perdedor 0 puntos. Asuma que no hay empates.

- Para este laboratorio, la manera en la cual se decidirá quién ganará en cada partida es comparando los ratings de cada jugador (ver columna “rating” del .csv). Quien tiene mayor rating, gana.

- Al finalizar las 7 rondas, clasifica a la siguiente fase solo 1 jugador: aquel que tenga mayor puntaje.

Fase final

El ganador de la fase de rondas se enfrenta con el campeón vigente, para este laboratorio lo llamaremos “Anand” (el campeón del torneo del año anterior).

Fase final

Ganador de fase de rondas	vs	Anand
---------------------------	----	-------

En la fase final los 2 finalistas jugarán 12 partidas, y el que obtiene el mejor puntaje al final de las 12 partidas gana el campeonato.

En esta fase los resultados posibles por cada partida para cada jugador son: ganar (1 punto), empatar (0.5 punto), perder (0 puntos).

Los resultados de cada partida en esta fase van a ser determinados de manera aleatoria.

Duración de cada partida: 0.15 segundos

- a) Leer el contenido del archivo 'players.csv' **(1 pto)**
- b) Desarrollar la función *fase_rondas_async()* que implementa la fase de rondas de manera asíncrona. Esta función debe retornar el nombre del único jugador que pasará a la siguiente fase. Nota: Deberá asumir que los 2 partidos de cada ronda se juegan en simultáneo. **(4 ptos)**
- c) Desarrollar la función *fase_rondas_sync()* que hace lo mismo que la parte b) pero de manera síncrona. **(2 ptos)**
- d) Desarrollar la función *fase_final_async()* que implementa la fase final de manera asíncrona. Esta función debe retornar el nombre del campeón del torneo. Nota: En este ítem asuma que las 12 partidas se van a jugar en simultáneo. **(4 ptos)**
- e) Desarrollar la función *fase_final_sync()* que calculará los resultados de la fase final de manera síncrona con un cambio en las reglas: el ganador es aquel que llega primero a obtener 6.5 puntos; es decir, ya no necesariamente van a jugar las 12 partidas. El resto de reglas se mantiene. **(3 ptos)**
- f) Mida los tiempos de ejecución de la parte b) y c). Explique de manera teórica a qué se debe la diferencia entre los tiempos de ejecución **(1 pts)**

Experiencia 2: Uso de threads

En el archivo [experiencia2_plantilla.py](#) se ha definido la función `get_ntp_time()` la cual imprime la hora en un país determinado. Su argumento de entrada es cualquiera de las cadenas definidas en la lista `servidores_ntp[]`. Lo que hace internamente la función es obtener la hora de esas URL según el país solicitado. Por ejemplo:

Ejemplo1: si Ud. llama a la función de esta manera:

`get_ntp_time("1.es.pool.ntp.org");` la función imprimirá la hora actual en España.

Ejemplo2: si Ud. llama a la función de esta manera:

`get_ntp_time("0.br.pool.ntp.org");` la función imprimirá la hora actual en Brasil.

No tiene que modificar `get_ntp_time()` por dentro, tampoco es necesario que entienda cómo es que funciona, solo úsela para lo siguiente:

a. **(2 puntos)** Escriba una función que itere sobre todas las cadenas en la lista `servidores_ntp[]` para que imprima la hora de todos los países. Imprima el tiempo de ejecución de esta función. Nota: Esos servidores son gratuitos y usados por muchas personas, puede que a veces alguno de ellos no responda. Si Ud. ve que alguna URL se está demorando más de 10 segundos en imprimir su hora,

cancela la ejecución de su programa(ctrl+C) y vuélvalo a ejecutar. Si el problema persiste, puede quitar esa URL de la lista `servidores_ntp[]`.

b.(3 puntos) Haga lo mismo que en la parte a) pero en vez de iterar sobre la lista, use `threads` para cada uno de los elementos en la lista `servidores_ntp[]`. Imprima el tiempo de ejecución de esta función.

Muestre screenshots con los tiempos de ejecución de ambas funciones. ¿Usar `threads` hace más rápido el programa?