

## 1IEE14 - Laboratorio 10

### Instrucciones para el laboratorio:

- Materiales permitidos: Wiki del curso, apuntes de clase, consultar foros, tutoriales o documentación de python online.
- Está prohibido el uso de cualquier modelo de lenguaje como ChatGPT o alguno similar. A cualquier alumno que se le detecte que ha consultado un modelo de lenguaje se le pondrá nota 0(cero) en el laboratorio.
- Usted debe subir a Paideia 1 solo archivo comprimido(.zip o .rar) con el nombre L10\_CODIGOPUCP.zip o L10\_CODIGOPUCP.rar. Este archivo comprimido debe tener archivos de python(extensión .py) para cada pregunta. No se aceptarán soluciones en Jupyter notebook.
- Se espera que el nombre de los archivos sea: pregunta1.py, pregunta2.py, etc.
- El horario máximo permitido para subir el archivo es a las 10:00:00 pm. Pasada esa hora, habrá una penalidad de 2 puntos por cada minuto extra que se demore en entregar su archivo.

1. **(1 punto)** Se quiere simular una subasta simple de un item valioso. En esta subasta simple, van a haber 5 participantes, cada uno de los cuales va a elegir un precio que están dispuestos a pagar por el item.

El vendedor cree que como mínimo debe recibir \$20000, por lo que ese será el precio base de la subasta(esto para prevenir que los participantes oferten solo \$1 o \$10 por el item).

Simule los 5 participantes con funciones asíncronas. Cada uno se va a demorar un tiempo aleatorio entre 0-10 segundos (use asyncio.sleep) para decidir cuál va a ser su oferta. Una vez que el sleep asíncrono acabe, use una función aleatoria para decidir el precio por cada participante. Al final, se deben comparar los precios ofrecidos y se debe imprimir quién ganó la subasta.

Este es un screenshot de cómo debería verse la ejecución de su programa:



```
"D:\Ciclo 2023-1\lab11_telecom_solus\venv\Scripts\python.exe" "D:\Ciclo 2023-1\lab11_telecom_solus\Scripts\python.exe"
Ofertas finales: {'b': 90315, 'd': 54191, 'a': 34472, 'e': 50841, 'c': 81600}
El ganador es: b

Process finished with exit code 0
```

Figura 1. Ejecución de la pregunta 1

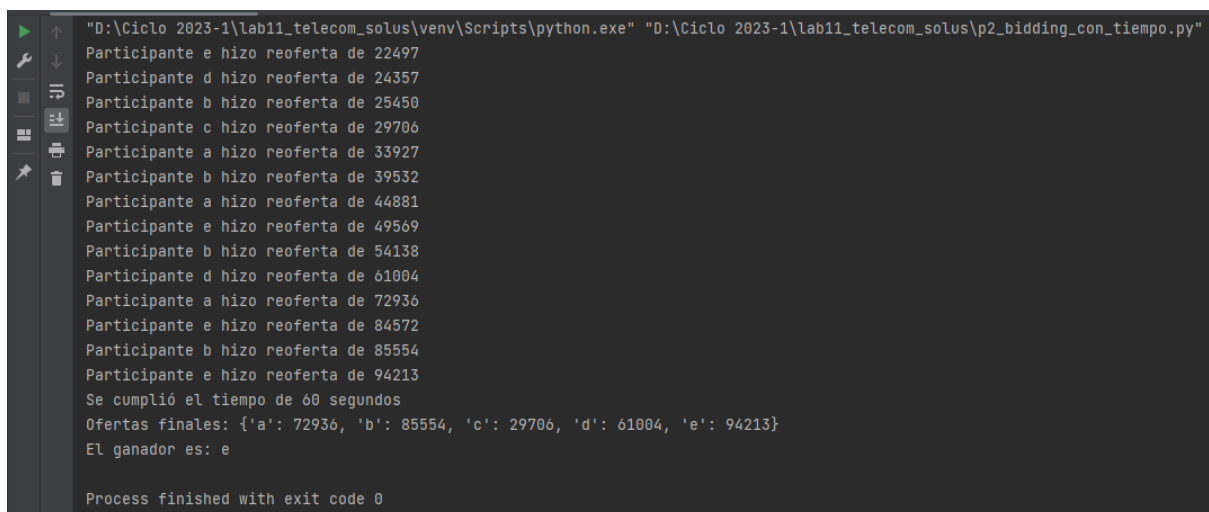
En el screenshot de arriba, a los participantes se les puso de nombre las letras “a”, “b”, “c”, “d”, “e”. Los números que se ven son sus ofertas. Cada uno se demoró un tiempo aleatorio usando `asyncio.sleep()` en decidir su precio. Al final el ganador fue “b” porque él ofreció el precio más alto por el item.

Nota: En el screenshot se observa que se ha usado un diccionario. No es obligatorio que Ud. lo haga así. Lo importante es que resuelva el problema usando `asyncio.sleep` para simular el tiempo que se demora cada participante en decidirse por un precio.

**2. (3 puntos)** En el ejercicio de la pregunta 1 se asumió de que cada participante no tiene la oportunidad de reofertar y que no tiene información del resto de participantes. Para este nuevo ejercicio asuma que la subasta se realiza de manera online y es temporizada. Estas son las reglas de la nueva subasta:

- Se seguirá usando un precio base de \$20 000
- La subasta durará 60 segundos
- Dentro de esos 60 segundos, cada participante hará una oferta(ayúdese de la librería `random` para generar montos aleatorios). Debe seguir haciendo uso de `asyncio.sleep` para simular un tiempo aleatorio que el participante se demora en tomar una decisión.
- Dentro de esos 60 segundos, cuando un participante hace una oferta, los otros chequean si este nuevo precio es mayor que el que ellos han ofrecido. Si lo es, ellos decidirán aleatoriamente si es que van a hacer una reoferta a un precio más alto. Si es que van a hacer una reoferta, el incremento mínimo debe ser de \$500 (esto para evitar que cada participante suba el precio en incrementos de \$1 o en centavos).
- Al finalizar los 60 segundos, el que haya ofrecido el precio más alto se lleva el item subastado.

Este es un screenshot de cómo debería verse la ejecución de su programa:




```
"D:\Ciclo 2023-1\lab11_telecom_solus\venv\Scripts\python.exe" "D:\Ciclo 2023-1\lab11_telecom_solus\p2_bidding_con_tiempo.py"
Participante e hizo reoferta de 22497
Participante d hizo reoferta de 24357
Participante b hizo reoferta de 25450
Participante c hizo reoferta de 29706
Participante a hizo reoferta de 33927
Participante b hizo reoferta de 39532
Participante a hizo reoferta de 44881
Participante e hizo reoferta de 49569
Participante b hizo reoferta de 54138
Participante d hizo reoferta de 61004
Participante a hizo reoferta de 72936
Participante e hizo reoferta de 84572
Participante b hizo reoferta de 85554
Participante e hizo reoferta de 94213
Se cumplió el tiempo de 60 segundos
Ofertas finales: {'a': 72936, 'b': 85554, 'c': 29706, 'd': 61004, 'e': 94213}
El ganador es: e
Process finished with exit code 0
```

Figura 2. Note cómo, a diferencia de la pregunta 1, ahora los participantes hacen reofertas si es que se dan cuenta que otro ha ofrecido más que ellos.

**3) (5 puntos)** En el ejercicio de la pregunta 2 se asumió de que los participantes no usan ningún tipo de estrategia a su favor. Una estrategia en subastas online es usar un “Sniper”, el cual es un programa informático que introduce una oferta alta faltando pocos segundos a que acabe la subasta. Como esta oferta se hace faltando pocos segundos, los otros participantes tienen poco tiempo a reaccionar y pierden la subasta. Copie y pegue su archivo de la pregunta 2 y cámbiele de nombre a `pregunta3.py`, y haga las siguientes modificaciones:

- Todas las reglas de la pregunta 2 se mantienen.
- Añada un sexto participante a su código el cual simulará a ser el Sniper.
- El Sniper pondrá una orden de oferta faltando solo 3 segundos antes de que se cumplan los 60 segundos.
- Entre los segundos 0 y 56, usted va a tener corriendo un programa cliente, el cual recibirá un número N por el terminal, se lo enviará a un programa servidor usando sockets, y el servidor lo guardará en un archivo llamado *oferta\_del\_sniper.txt*
- El precio que ofertará el Sniper en el Segundo 57 lo leerá del archivo *oferta\_del\_sniper.txt* y será el mismo número N que usted ingresó por el terminal en su programa cliente.

Este es un screenshot de cómo debería verse la ejecución de su programa:



```
Run: p3_bidding_con_sniper x
"D:\Ciclo 2023-1\lab11_telecom_solus\venv\Scripts\python.exe" "D:\Ciclo 2023-1\lab11_telecom_solus\p3_bidding_con_sniper.py"
Participante c hizo reoferta de 23954
Participante b hizo reoferta de 29084
Participante a hizo reoferta de 29895
Participante d hizo reoferta de 31864
Participante e hizo reoferta de 33903
Participante a hizo reoferta de 38534
Participante d hizo reoferta de 41692
Participante e hizo reoferta de 45200
Participante b hizo reoferta de 54483
Participante sniper hizo reoferta de 100000
Participante d hizo reoferta de 109860
Se cumplió el tiempo de los 60 segundos
Ofertas finales: {'a': 38534, 'b': 54483, 'c': 23954, 'd': 109860, 'e': 45200, 'sniper': 100000}
El ganador es: d

Process finished with exit code 0
```

Figura 3. Se agregó un participante más llamado sniper. En el screenshot, hizo una oferta de \$100 000 en el segundo 57. Note de que eso no es garantía de que gane, porque alguien puede ofrecer un precio más alto justo en el ultimo segundo.

**4) (6 puntos)** En una subasta del espectro radioeléctrico, el gobierno concede cierto ancho de banda a las empresas de telecomunicaciones como Telefónica, Claro, o Entel por cierta cantidad de años. Aquí lo que se subasta son varios bloques de anchos de banda a la vez. Por ejemplo:

Bloque 0: 50 MHz

Bloque 1: 60 MHz

Bloque 2: 70 MHz

El método más simple sería aplicar el tipo de subasta de la pregunta 1 por cada bloque. La desventaja de eso es que a las empresas les conviene comprar bloques que estén cercanos entre sí, por lo que no les convendría ganar un bloque pero perder el siguiente.

Para solucionar eso, un tipo de subasta es la SMRA (Simultaneous Multiple Round Ascending), la cual tiene estas reglas:

- Asuma solo 3 participantes: Telefónica, Claro, y Entel
- Se empieza con la ronda 1, la cual durará 30 segundos y sigue la misma dinámica que la pregunta 2; es decir, cada participante puede reofertar en esos 30 segundos. La diferencia con la pregunta 2 es que acá cualquier participante puede ofrecer un nuevo precio para cualquier número de bloques.
- Terminada la ronda 1, se imprime en consola los precios con los que quedaron el bloque 0, 1, y 2.
- Se empieza con la ronda 2 la cual empieza con su precio base igual a los precios finales de la ronda 1.
- El proceso se repite para la tercera y última ronda.

Finalizada la tercera ronda, se imprimen los ganadores de cada bloque.

Este es un ejemplo de cómo se desarrollaría una subasta de este tipo:

Bloques a subastar:

Bloque 0: 50 MHz

Bloque 1: 60 MHz

Bloque 2: 70 MHz

Precio base de cada bloque: \$ 15 millones

Ronda 1:

Precios actuales:

Bloque 0: \$15 millones

Bloque 1: \$15 millones

Bloque 2: \$15 millones

Participante Telefónica hizo oferta de 18 para el bloque 0

Participante Claro hizo oferta de 23 para el bloque 1 y 2

Participante Entel hizo oferta de 30 para el bloque 2

Participante Telefónica hizo oferta de 38 para el bloque 0 y 2

Se cumplió el tiempo de 30 segundos. Ronda 1 finaliza.

Ronda 2:

Precios actuales:

Bloque 0: \$38 millones

Bloque 1: \$23 millones

Bloque 2: \$38 millones

Participante Claro hizo oferta de 25 para el bloque 1

Participante Telefónica hizo oferta de 40 para el bloque 0

Participante Entel hizo oferta de 43 para el bloque 0, 1, y 2

Se cumplió el tiempo de 30 segundos. Ronda 2 finaliza.

Ronda 3:

Precios actuales:

Bloque 0: \$43 millones

Bloque 1: \$43 millones

Bloque 2: \$43 millones

Participante Claro hizo oferta de 45 para el bloque 0

Participante Entel hizo oferta de 48 para el bloque 0

Participante Telefónica hizo oferta de 50 para el bloque 0 y 1

Se cumplió el tiempo de 30 segundos. Ronda 3 finaliza.

**Los ganadores son:**

Bloque 0: Telefónica con \$50 millones

Bloque 1: Telefónica con \$50 millones

Bloque 2: Entel con \$43 millones