1IEE14 - Laboratorio 11

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 L11_CODIGOPUCP.zip o L11_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. A no ser que la pregunta especifique un nombre para su(s) archivo(s).
- 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.

Pregunta 1 **(10 puntos)**

Se solicitará dos archivos: “cliente.py” y “servidor.py”.

a. (2 puntos) Cree un primer thread en el archivo servidor.py el cual, luego de establecer conexión con un cliente, leerá el archivo PartesDeElectrónica.csv

La cabecera de este archivo es: [ID, Nombre, Número de parte, Cantidades, Peso, Costo unitario]

Este archivo contiene los datos de varios componentes electrónicos que se quieren comprar en un proveedor de China.

b. (2 puntos) El segundo Thread del servidor, en intervalos de 1 segundo, le enviará al cliente la cadena de caracteres de una fila (datos de un componente electrónico), así consecutivamente hasta completar de leer el archivo.

c. (1 punto) En el archivo cliente.py, cree un thread que se encargue de recibir los datos del servidor.

d) (3 puntos) Cree un segundo thread en el cliente que se encargue de procesar los datos de cada fila recibida. Debe calcular lo siguiente:

- Cálculo del costo total (CT = Precio unitario * cantidades)

- Clasificación del componente por costo total:

  (Menor a 25): Costo bajo

  (25.0 – 49.9): Costo regular

  (50 - 74.9): Costo alto

  (75 a más): Costo elevado

- Conteo de componentes que tengan costo elevado, conteo de componentes con peso mayor a 100g y con costo elevado, y conteo de componentes con costo bajo (conteos actualizados por cada componente que se procese).

e) (2 puntos) Asimismo, el segundo thread imprimirá en terminal los resultados de cada fila de la siguiente forma:

```
--------------Nombre: XXXX-----------------
Costo total: XXX
Clasificación por costo: XXXX
Número de componentes con costo elevado: XXX
Número de componentes con costo elevado y con peso mayor a 100g: XXX
Número de componentes con costo bajo: XXX
--------------Nombre: XXXX-----------------
Costo total: XXX
Clasificación por costo: XXXX
Número de componentes con costo elevado: XXX
Número de componentes con costo elevado y con peso mayor a 100g: XXX
Número de componentes con costo bajo: XXX
--------------Nombre: XXXX-----------------
Costo total: XXX
Clasificación por costo: XXXX
Número de componentes con costo elevado: XXX
Número de componentes con costo elevado y con peso mayor a 100g: XXX
Número de componentes con costo bajo: XXX
```

Nota: solo debe imprimir en terminal si y solo si se tiene un nuevo dato a imprimir

Pregunta 2 **(5 puntos)**

En la plantilla pregunta2_lab11_h621_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.uk.pool.ntp.org"); la función imprimirá la hora actual en Reino Unido.

Indicaciones sobre la función get_ntp_time():

- Aparte de imprimir la fecha-hora, también retorna el timestamp en formato datetime.datetime.
- Los países disponibles en la plantilla son aquellos donde se encuentran las principales bolsas de valores del mundo donde uno puede comprar o vender acciones, lo cual será importante para este ejercicio. 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 determinar cuál es el país cuya bolsa de valores está más próxima a abrir con respecto a nosotros(en Perú). Asuma que todas las bolsas de valores abren a las 08:00am en sus respectivos países.

Por ejemplo: Supongamos que en actualmente en Perú fueran las 12:00, y en los otros países las horas son:

- UK: 17:00

- España: 18:00

- Estados Unidos: 12:00

- Hong Kong: 01:00

- Japón: 02:00

Entonces la respuesta correcta es que Japón sería el más próximo a abrir, porque es el que está más cercado a las 08:00am (pues solo le faltan 6 horas para abrir). Su programa al final **debe** imprimir el país que ha calculado tener la hora más cercana. Imprima el tiempo de ejecución de esta función.

Nota: Esos servidores son gratuitos y usan un tipo de protocolo que no responde el 100% de las veces, así que 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, cancele 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[]. Una vez terminado los threads, su programa debe imprimir el país que ha calculado tener la hora más cercana. Imprima el tiempo de ejecución de esta función. Indique cuál función fue más rápida, si la implementada en la parte a) o en la parte b) en base a sus tiempos de ejecución