

## 1IEE06 - 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.
- Si al momento de ejecutar cualquiera de sus programas le sale el error *Address already in use*, eso se debe a que el puerto que usaba sigue abierto y no está disponible para su uso. Cuando ocurra esto, puede cambiar de puerto, no es obligatorio usar solamente el puerto 5000.
- Está prohibido usar cualquier librería como ayuda para leer o escribir en los archivos .csv
- 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 (3 puntos)

- a. Cree un archivo llamado *p1\_servidor.py* y escriba una función en Python que recibe como parámetro de entrada un número entero *x* y retorna el factorial de *x*. (1.0 punto)
- b. En el mismo archivo *p1\_servidor.py*, escribir un servidor TCP que reciba un número entero *n* vía socket y le entregue al cliente el factorial del número. (1.0 punto)
- c. Cree un archivo llamado *p1\_cliente.py* y escriba un cliente que reciba un número *n* vía el terminal, se lo envía a *p1\_servidor.py*, recibe el resultado del factorial calculado por el servidor y lo imprime en el terminal. (1.0 punto)

### Pregunta 2 (7 puntos)

- Implemente el juego Ahorcado en Python. Para implementarlo, tomar en cuenta lo siguiente:
- Se va a tener 1 jugador(el cliente), el cual va a interactuar con el servidor vía sockets para intentar adivinar la palabra.
- Para empezar el juego, el cliente debe enviar el mensaje 'start'.
- Una vez empezado el juego, el cliente tiene que adivinar la palabra correcta enviando 1 caracter a la vez.

- Cada cliente solo se puede equivocar como máximo en 5 oportunidades. El cliente ya está hecho, descárguelo aquí: [cliente\\_ahorcado.py](#)
- Cuando el juego acaba, ya sea porque el cliente adivinó la palabra o perdió, queda a su criterio decidir si es que el juego vuelve a solicitar el inicio de un nuevo juego, o si el programa finaliza su ejecución. Ambas decisiones son válidas.

Ud. tiene que implementar solamente el servidor. Use el archivo [servidor\\_ahorcado\\_plantilla.py](#) como ayuda.

El archivo [funcion\\_reemplaza\\_caracteres.py](#) le provee una función de ayuda para el juego.

Así es como debe verse el juego:

```

Terminal: Local x + v
(venv) PS D:\Ciclo 2023-2\labs_telecom> python .\p3_cliente_ahorcado.py
Conectandome a servidor....
¡Conectado!
Típee lo que quieres enviar:
start
Mensaje recibido: b'****'
Tienes 5 oportunidades!
Típee lo que quieres enviar:
u
Mensaje recibido: b'*u**'
Típee lo que quieres enviar:
a
Mensaje recibido: b'Intento incorrecto'
Típee lo que quieres enviar:
p
Mensaje recibido: b'puxp'
Típee lo que quieres enviar:
c
Mensaje recibido: b'pucp es la respuesta. Ganaste!'
Típee lo que quieres enviar:
[]

(venv) PS D:\Ciclo 2023-2\labs_telecom> python .\p3_servidor_ahorcado_solu.py
Arrancando servidor en 127.0.0.1:5900
Esperando una nueva conexion de un cliente
...conexion de: ('127.0.0.1', 52862)
recbi comando start
Palabra elegida: pucp
Client guess: u
Client guess: a
Client guess: p
Client guess: c
Closing connection
(venv) PS D:\Ciclo 2023-2\labs_telecom>

```

### **Pregunta 3 (6 puntos)**

Codificar dos scripts (p3\_cliente.py – p3\_servidor.py). El cliente debe leer un archivo **.csv** que contenga la información de las notas de 4 prácticas, 10 laboratorios, un examen parcial y un examen final. Todas las notas deben estar en una sola fila separadas por comas.

El cliente debe leer las notas y las debe enviar al servidor el cual debe responder con el cálculo del promedio final siguiendo la ecuación:

$$NF = (3*Pa + 3*Pb + 4*Ta) / 10$$

Donde Pa es el promedio de prácticas sin considerar la menor nota, Pb es el promedio de laboratorios sin considerar las 2 notas más bajas y Ta es una tarea académica. Además, debe considerar lo siguiente:

- Para hallar el mínimo valor de una lista puede utilizar `min` de `numpy`.
- Para hallar el promedio de una lista puede utilizar `mean` de `numpy`.
- Para remover un elemento de una lista puede hacer uso del comando `remove` (`lista.remove(elemento)`)

#### **Pregunta 4 (4 puntos)**

La base de datos de un hospital tiene alojada la información sobre pacientes con una determinada enfermedad. Para ello, se sabe que el archivo que contiene la base de datos es un archivo con extensión `.csv` el cual está compuesto de las columnas paciente, edad y el diagnóstico (0 sano, 1 enfermo). Se le pide:

- Crear una pequeña base con 10 pacientes. En la columna pacientes considerar del 1 al 10. Edades aleatorias y diagnóstico aleatorio.
- Crear un script servidor(`p4_servidor.py`) que lea la información del archivo `.csv`. El servidor analizará lo que el cliente le pida y según ello le enviará la información requerida.
- Crear un script cliente(`p4_cliente.py`). Este script solo puede pedir datos estadísticos una sola vez. Sus opciones son pedir:
  - el promedio de edades de los pacientes
  - el número de pacientes enfermos
  - el número de pacientes sanos.

El cliente enviará solamente la letra 'a', 'b', o 'c'. El servidor debe saber cómo interpretar cada opción.

Finalmente, el cliente debe guardar la información recibida en un archivo **`.csv`**