

## 1IEE14 - Laboratorio 7

### 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 usar Github CoPilot. A cualquier alumno que se le detecte que ha consultado un modelo de lenguaje se le pondrá nota 0(cero) en el laboratorio.
- No está permitido el uso de la librería Pandas ni ninguna otra librería para leer y/o escribir en los archivos csv.
- Usted debe subir a Paideia 1 solo archivo comprimido (.zip o .rar) con el nombre L7\_CODIGOPUCP.zip o L7\_CODIGOPUCP.rar. Este archivo comprimido debe tener archivos de python para cada pregunta. Note que se le está pidiendo archivos de python para cada pregunta, no se aceptarán soluciones en jupyter notebook.
- 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** (5 puntos)

Cree un archivo llamado **pregunta1.py** y dentro del archivo escriba una función que lea el archivo `pacientes.csv` (está en Paideia) hasta la línea  $N$ , ordene a los pacientes por edad y lo guarde en otro archivo llamado `pacientes_ordenado.csv`.

- a. Realice una función que realice lo indicado y devuelva una lista con los tiempos de lectura, escritura y el tiempo que toma ordenar los pacientes para el  $N$  que se le ingresa. **(4 puntos)**
- b. Para los siguientes valores de  $N = 500, 2500, 5000$  imprima los valores lecturas de lectura, escritura, y el tiempo que toma ordenar los pacientes. **(1 punto)**
- Al ejecutar su programa, se espera que imprima lo siguiente:

Para  $N = 500$ , el tiempo de lectura es xxx, el de escritura es yyy, y el de ordenar los pacientes es zzz.

Para  $N = 2500$ , el tiempo de lectura es xxx, el de escritura es yyy, y el de ordenar los pacientes es zzz.

Para  $N = 5000$ , el tiempo de lectura es xxx, el de escritura es yyy, y el de ordenar los pacientes es zzz.

## **Pregunta 2** (10 puntos)

Se tiene a disposición 50 archivos de tipos .csv(están en Paideia) de los cuales cada uno contiene 80000 filas y 8 columnas, donde cada celda tiene asignado un número aleatorio del 0 al 9.

a. En el archivo **pregunta2a.py**, escriba una función que reciba como parámetro de entrada su código de alumno y halle la cantidad de veces que este se encuentra en los archivos de forma horizontal. Es decir, debe revisar en cada fila de cada uno de los archivos si su código hace “match”. Luego debe mostrar un histograma que indique la cantidad de veces que hubo match en cada archivo, donde el eje X corresponderá a cada archivo y el eje Y a la cantidad de “matches”. Este histograma debe guardarse como imagen de la forma **“hist2\_a.png”** (4 puntos)

b. En el archivo **pregunta2b.py**, escriba una función similar al de la pregunta 2a, pero esta vez que busque su código en cada columna(de manera vertical), tenga en cuenta que en una sola columna solo debe revisar de 8 en 8, es decir, que en una columna de N números, debe sólo realizar un análisis N/8 veces. Asimismo, el histograma debe tener el formato **“hist2\_b.png”**. (3 puntos)

c. Para el inciso (a) y (b) evalúe los tiempos de ejecución, de forma separada. Es decir, edite cada uno de los códigos para que se imprima en terminal el tiempo de ejecución del programa. Comente acerca de sus resultados. (1 puntos)

d. En el archivo **pregunta2d.py**, haga lo mismo que la pregunta 2a (búsqueda for filas), pero esta vez haciendo lecturas parciales: Vaya leyendo los archivos en grupos de 500 filas. Mida el tiempo de ejecución de esta función y compárelo con el tiempo que obtuvo en 2c. Comente si este método de lecturas parciales es más rápido o más lento que el método de leer cada archivo por complete. (2 puntos)

**A tomar en cuenta al momento de subir sus archivos:** Para la pregunta 2, aparte de incluir sus archivos de Python, debe incluir sus archivos .png también.

**Sugerencia:** Puede usar el siguiente fragmento de código para graficar el histograma:

```
import matplotlib.pyplot as plt
datos_histograma = []
plt.hist(datos_histograma)
plt.savefig('hist2_a.png')
```