

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

**ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS**

**Laboratorio 4  
(Segundo semestre 2023)**

**Indicaciones generales:**

- Debe enviar su solución, siguiendo el formato indicado en cada pregunta.
- **La hora de entrega es hasta culminadas las 2 horas de evaluación de 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) con el nombre L10\_CODIGO.zip. Este archivo comprimido debe tener archivos de python(extensión .py) para cada pregunta.

**Experiencia 1 (10 puntos)**

Se tiene información de las ciudades importantes del Perú y las capitales de Sudamérica (nombre, latitud y longitud) en un archivo csv denominado `coordenadas.csv` en un servidor. Se le pide desarrollar una comunicación por sockets con un cliente que solicite las N ciudades más cercanas a la ciudad que se ingresa.

**`cliente_1.py`**

- Permite a un usuario ingresar el número de ciudades más cercanas N junto con el nombre de la ciudad a través del terminal.
- Realiza la solicitud al servidor.
- Guarda el nombre de la ciudad y la distancia a esta en un archivo csv (`ciudades.csv`).
- Cierra la conexión

**`servidor_1.py`**

- Recibe la solicitud de un cliente.
- Lee el archivo csv.
- Envía los datos sin modificar el archivo.
- Cierra la conexión con el cliente.

Mida el tiempo de lectura, envío, recepción y escritura, en el cliente o en el servidor, según corresponda. Mostrar el tiempo en milisegundos.

## Ejemplo de ejecución

```
$ python3 servidor_1.py
El tiempo de ??? es: x ms
El tiempo de ??? es: x ms
```

```
$ python3 cliente_1.py
Ingresa una ciudad: Lima
Ingresa el número N: 10
El tiempo de ??? es: x ms
El tiempo de ??? es: x ms
```

En el archivo `ciudades.csv`:

```
Ciudades,Distancia
Cerro de Pasco,1.7e+02
Huancaayo,2e+02
Huanuco,2.5e+02
Ica,2.7e+02
Huaraz,2.8e+02
Ayacucho,3.3e+02
Chimbote,3.7e+02
Trujillo,4.9e+02
Cajamarca,5.7e+02
Cusco,5.7e+02
```

Para calcular las distancias entre dos ciudades es necesario utilizar la fórmula de Haversine:

$R = 6,371 \text{ km}$  (radio de la Tierra)

$\Delta lat = lat_2 - lat_1$

$\Delta long = long_2 - long_1$

$a = \sin^2(\Delta lat/2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2(\Delta long/2)$

$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$

$d = R \cdot c$

## **Experiencia 2 (10 puntos)**

Se desea enviar el archivo **binario** `data.rf` a través de una conexión por sockets. En el cliente, quien es el que descarga el archivo, el tamaño de buffer debe ser de N bytes y el archivo descargado se debe llamar `descarga.rf`

- a) (1pt) Realizar los scripts denominados `servidor_2.py` y `cliente_2a.py` para un N genérico. La descarga debe empezar cuando el cliente establece la conexión. Mida el tiempo de lectura, envío, recepción y escritura, en el cliente o servidor, según corresponda.
- b) (2pt) Medir el tiempo total de envío y recepción del archivo para N = 8, 16, 32 ... , 1024. **No considerar el tiempo de lectura o escritura del archivo en disco.** Guardar todos los tiempos en un arreglo denominado `tiempos_N`. Utilizar el siguiente código para generar una gráfica de Tamaño de paquete vs tiempos de ejecución:

```
import matplotlib.pyplot as plt
import socket
import time

lista_N = [8,16,32,64,128,256,512,1024]
tiempos_N = []

# Inserte aqui el codigo que haga falta

for N in lista_N:
    # Inserte aqui el codigo que haga falta
    tiempos_N.append(tiempo_medido)

plt.plot(lista_N,tiempos_N)
plt.xlabel('Packet size [bytes]')
plt.ylabel('Transfer time [ms]')
plt.savefig('Recepcion.png') # Cambiar nombre segun
corresponda
```

- c) (2pt) ¿Cómo afecta el tamaño del paquete al tiempo de envío y recepción del archivo? Responda en un comentario tanto en el servidor como en el cliente.
- d) (2pt) Realizar el script denominado `cliente_2b.py`, en donde se intercale la recepción y escritura del archivo en paquetes de N bytes. Las siguientes dos preguntas deben responderse como comentarios en el script `cliente_2b.py`:
  - e) (2pt) ¿La implementación en d) es más o menos eficiente que la implementación en a) en términos de tiempo de ejecución? ¿Por qué?
  - f) (1pt) ¿Cuál alternativa requiere más RAM? ¿Por qué?