

## Proyecto 1

### Implementación Sockets

**Profesor:** Cesár Azurdia M.

**Auxiliares:** Sandy Bolufe, Alejandro Cuevas.

**Ayudante:** Nicolás Ortega

Quizás la aplicación más conocida de los *sockets* TCP/IP es el chat. En esta tarea se tiene por objetivo crear un chat simplificado, donde los mensajes solo se mostraran en la ventana del servidor. Para esto, se puede utilizar el código base de Python ubicado en material docente, sin embargo usted es libre de realizar la tarea como desee.

Este sistema de chat debe cumplir con ciertos requerimientos, los cuales son:

1. El servidor debe ser capaz de aceptar al menos 6 clientes distintos.
2. Cada cliente debe tener un identificador único <sup>1</sup>
3. Los mensajes recibidos deben ser mostrados en la consola del servidor, siguiendo el formato:

```
<Identifier>: Message
```

4. Cuando se conecte un cliente, debe mostrar un mensaje tanto en la consola del servidor avisando que se conectó un cliente nuevo con su respectivo identificador como en la consola del cliente avisando que fue exitosa su conexión al chat, el mensaje debe ser enviado desde el servidor. Un ejemplo de la salida puede ser:

#### Terminal servidor

```
<Identifier1>: Message 1
<Identifier1>: Message 2
[SERVER] Client: <Identifier2> connected!
<Identifier2>: Message 3
<Identifier2>: Message 4
```

#### Terminal cliente

```
Welcome to the best chat in the universe!
```

5. Cree un comando de escape del chat, en este caso cuando el cliente ingrese :q, es decir, si el cliente envía dicho string el socket respectivo debe ser cerrado.
6. La implementación debe ser en base a *threads*, donde se debe levantar un hilo por cada conexión nueva, en cada hilo debe ejecutarse una función que primero salude al cliente, muestre en la consola que se conectó un cliente nuevo y se mantenga recibiendo los mensajes hasta que se termine la conexión. <sup>2</sup>.

<sup>1</sup>En Python, puede usar el método de la librería socket `getpeername()` o puede enviar una cabecera que solo deje conectar si es única.

<sup>2</sup>En Python, puede usar la librería *threading*



Una vez implementado el chat, usted debe entregar un reporte que contenga:

1. Introducción.
2. Marco teórico que debe al menos responder:
  - ¿Qué es un socket?
  - ¿Cómo funciona un socket?
  - ¿Qué tipo de socket utilizó en la implementación y cuáles son sus características?
3. Descripción del servidor: Describir la implementación que realizó. Una buena forma para esto es explicar las partes relevantes del código mostrando partes del código mismo.
4. Discusión y conclusiones.

**Además deberá entregar los códigos implementados correctamente comentados.**