



**UNIVERSIDAD LATINA  
DE COSTA RICA**

POWERED BY **Arizona State University**

## **Sistemas Operativos II**

### **Profesor**

Carlos Mendez Rodriguez

### **Estudiante:**

Javier Díaz Mora / 20200120139

## Socket

- **Socket:** Es un punto final en una comunicación bidireccional en una red. En TCP un socket es una combinación de una dirección IP y un número de puerto. TCP (Transmission Control Protocol) que es un protocolo de nivel de transporte que asegura la entrega ordenada y confiable de datos entre dos sistemas.
- **Creación del socket:** `socket.socket(socket.AF_INET, socket.SOCK_STREAM)` lo que hace es que se configura un socket para utilizar IPv4 (AF\_INET) y TCP (SOCK\_STREAM).

## Threading

- **Threading** Se utiliza para manejar múltiples conexiones al mismo tiempo. En lugar de que el servidor se bloquee mientras maneja una conexión, cada conexión de cliente se maneja en un hilo separado, permitiendo que el servidor continúe aceptando nuevas conexiones.

## Interacción entre Sockets y Threading

- Cada hilo en el servidor tiene su propio socket de cliente, lo que le permite enviar y recibir mensajes de ese cliente. Los mensajes de los clientes pueden ser retransmitidos a otros clientes a través de sus respectivos hilos y sockets.
- **Sincronización:** Aunque los hilos permiten la concurrencia, también pueden llevar a problemas de sincronización, como condiciones de carrera. En Python, puedes utilizar mecanismos como Locks para evitar que varios hilos accedan a recursos compartidos al mismo tiempo de manera no segura.

El servidor funciona en la Capa 4 (TCP) y la Capa 7, mientras que las capas inferiores (1-3) se encargan de mover los bits y paquetes entre los dispositivos conectados.

