**Sockets, ServerSocket**

**1. ¿Qué son los Sockets y los ServerSockets?**

* **Socket**: Es un punto final para enviar y recibir datos en una conexión de red. Es como el "enchufe" por donde pasa la información entre dos computadoras en una red.
* **ServerSocket**: Este se utiliza en el lado del servidor. Crea un "puerto" que está abierto y esperando que un cliente se conecte. Una vez que un cliente se conecta, se establece una comunicación.

**2. ¿Para qué sirven?**

Los **sockets** permiten la comunicación entre dispositivos. Son esenciales para que un programa (como un chat, una página web o una aplicación) pueda enviar y recibir información a través de una red (como Internet).

* **Socket** (cliente): Enlaza al programa con la IP y el puerto del servidor al que quiere conectarse.
* **ServerSocket** (servidor): Escucha solicitudes de los clientes y acepta conexiones.

**3. ¿Cómo se utilizan?**

El flujo de uso típico es:

1. **Servidor**: Crea un ServerSocket, espera conexiones entrantes y, cuando una llega, se comunica con el cliente a través de un Socket.
2. **Cliente**: Crea un Socket que se conecta a un servidor para enviar y recibir información.

**4. ¿Cómo creo un cliente/servidor?**

Para crear una comunicación entre un cliente y un servidor, primero se configura el **servidor** para que esté disponible para aceptar conexiones de clientes. Esto se hace con un **ServerSocket**, que es como una puerta de entrada que espera que alguien toque para conectarse.

1. **Servidor**:
   * El servidor abre un **ServerSocket** en un puerto específico (por ejemplo, el puerto 5000) y se queda esperando a que un cliente se conecte.
   * Una vez que un cliente se conecta, el servidor acepta esa conexión y empieza a comunicarse con él a través de un **Socket**, que es la conexión establecida.
2. **Cliente**:
   * El cliente necesita conectarse al servidor. Para hacerlo, crea un **Socket** que apunta a la dirección IP del servidor y el puerto donde está escuchando.
   * Una vez que el cliente se conecta, ambos pueden enviarse y recibir datos.

El cliente y el servidor intercambian mensajes hasta que cierran la conexión. Así, ambos pueden comunicarse a través de una red usando sockets.