**Instituto Politécnico Nacional**

Escuela Superior de Cómputo

Reporte Práctica 3

**Integrantes del Equipo:**

Barrera Pérez Carlos Tonatihu

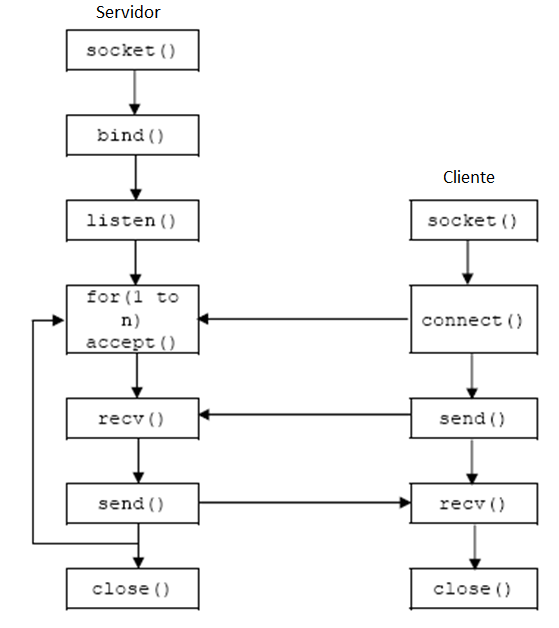
Castillo Reyes Juan Daniel

**Grupo:** 2CM11

**Unidad de Aprendizaje:** Aplicaciones para Comunicaciones en Red

**Profesor:** Axel Ernesto Moreno Cervantes

Introducción



**Sockets TCP**

Una conexión TCP es un canal abstracto bidireccional cuyos extremos están identificados por una dirección IP y un número de puerto. Antes de iniciar una comunicación una conexión TCP debe comenzar por el cliente TCP enviando una solicitud de conexión al servidor. En el caso de Java una instancia de la clase *ServerSocket* es la encargada de escuchar estas solicitudes de conexión y crear una nueva instancia de la clase *Socket* para manejar esta conexión.

**Cliente**

Los pasos que realiza el cliente para iniciar una conexión en el caso de Java son los siguientes:

1. Crear una instancia de la clase Socket con esto se establece una conexión con un host y un puerto.
2. Comunicarse usando flujos de entrada/salida (*InputStream/OutputStream*)
3. Finalmente, cerrar la conexión con el servidor *close()*.

**Servidor**

El objetivo del servidor es configurar la comunicación y esperar a los clientes. La forma en la que lo hace es la siguiente:

1. Crear una instancia de *ServerSocket* mediante la especificación de un puerto. Al hacer esto esperaremos por conexiones al puerto especificado.
2. Creamos una instancia de Socket al usar el método *accept()* de *ServerSocket*, nos comunicamos con el cliente mediante los flujos de entrada/salida. Terminamos la conexión con el cliente al llamar el método *close()* de *Socket*.
3. Repetimos el paso anterior las veces que sean necearías.

Desarrollo

Pruebas

Conclusiones

Referencias

* [1] K. Calvert and M. Donahoo, TCP/IP sockets in Java. San Francisco: Morgan Kaufmann Publishers, 2002.