



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA INSTITUTO DE COMPUTACIÓN

REDES DE COMPUTADORAS II

Tarea 1 – Evaluación 2

Sockets

Fecha: 02 de octubre de 2025

Grupo: 702-A

Objetivo: Que alumno refuerce el concepto de socket realizando una implementación del mismo.

Introducción

Socket es un mecanismo de comunicación punto a punto entre dos procesos sobre el protocolo TCP/IP. Se usa para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos requeridos. Existen dos tipos de comunicación: la orientada a la conexión y la no orientada a la conexión. En el primer caso se debe establecer la conexión entre ambos procesos mediante un socket para poder transmitir utilizando el protocolo TCP; en el segundo caso se utiliza el protocolo UDP, que garantiza que los datos que llegan son correctos pero no que lleguen todos.

En el contexto de programación, un socket es un archivo que se abre de una manera especial, donde se puede leer y escribir con las funciones específicas para tal funcionalidad. Dos de los lenguajes de programación más usados para programar un socket son: C, Python y Java.

Recursos necesarios

- Computadora con sistema operativo que soporte C, Python o Java según el lenguaje que se desee ocupar.
- Recomendaciones:
 - Para programar en C usar los paquetes `<sys/types.h>` y `<sys/socket.h>` principalmente.
 - Para programar en Python usar la librería `socket` y la función `socket.socket()` junto con otros métodos relacionados.
 - Para programar en Java la versión 5.0 o superior usando los paquetes: `java.io` y `java.net`

Actividad

Realizar una sesión con sockets TCP (o UDP), donde uno de los procesos será para representar el servidor que se encontrará en estado de escucha a la espera de recibir una conexión del segundo proceso que represente al cliente. Para realizar una conexión entre un cliente y un servidor usando sockets generalmente se debe de conocer la dirección IP de la máquina servidor y el puerto que identifica el servicio al que se quiere conectar.

El cliente y el servidor usarán métodos/funciones distintas; el esquema básico consta en que el servidor ocupa un papel pasivo y esperará solicitud(es) de conexión(es) posibles por parte del cliente; en tanto que el cliente será el encargado de solicitar conexión al servidor de forma activa.

Se deberá crear un socket de comunicación, una vez entablada la comunicación entre el cliente y el servidor, el cliente solicitará se le envíe una copia de un archivo que se encontrará en el servidor (cualquier tipo de formato: 1 texto(ASCCI/UNICODE), 2 tipo documento: pdf, word; 3 imágenes y 4 video multimedia). En cada una de las etapas de comunicación (desde la creación del socket hasta el cierre de sesión) el cliente deberá ser notificado de estas actividades mediante mensajes, incluyendo el mensaje de falla en caso de no conexión o no encontrarse en archivo solicitado.

No es necesaria una interface gráfica, sin embargo quien la entregue tendrá un puntaje extra(20%). Del mismo modo si no se cubren los requisitos indicados se penalizará. Debe considerar que se verificará que la modificación o el archivo haya sido exitoso, por ejemplo en el caso del archivo copiado, se revisará que el contenido sea el mismo que el que está en el origen.

Entrega

Viernes 10 de octubre a las 8:00am.

Subir un archivo pdf en la plataforma de Classroom, que incluya los siguientes datos:

- Portada, que incluya el nombre de los integrantes (equipo de 3 personas).
- El o los programas que incluyan las tareas del cliente y servidor, en el código se deberán colocar comentarios donde se encuentren las instrucciones donde se realizan las etapas de comunicación (por ejemplo: función de escucha, función de creación el socket, etc.).
- Presentar las conclusiones después obtenidas al realizar la práctica.

La revisión y pruebas se harán el mismo día 10 en la hora de clase, a 6 equipos. Ese día llevar su equipo de cómputo correspondiente o asegurarse que en la computadora a utilizar de la sala de cómputo tenga todo lo necesario para la ejecución de su(s) programa(s).