

PRACTICA PROGRAMACION EN RED: SOCKETS

**CAMILA BARONA
CAMILA MEJIA**

**UNIVERSIDAD EAFIT
TELEMÁTICA
2020**

INTRODUCCIÓN

El mecanismo socket es un protocolo de comunicación entre un programa del cliente y un programa del servidor en una red, es así, el punto final de una conexión.

Un socket está definido por un par de direcciones IP locales y remotas, un protocolo de transporte orientado a la conexión en el caso del proyecto e información de puertos locales y remotos.

Este proceso funciona como la comunicación por correo o por teléfono, donde de un buzón se extraen mensajes completos, mientras que el teléfono permite el envío de flujos de información que no tiene una estructura claramente definida hacia el cliente.

Así pues, es un proceso existente en la máquina cliente y en la máquina servidor, con el fin de que ambas puedan leer y escribir información, para luego ser retransmitida por las diferentes capas de red.

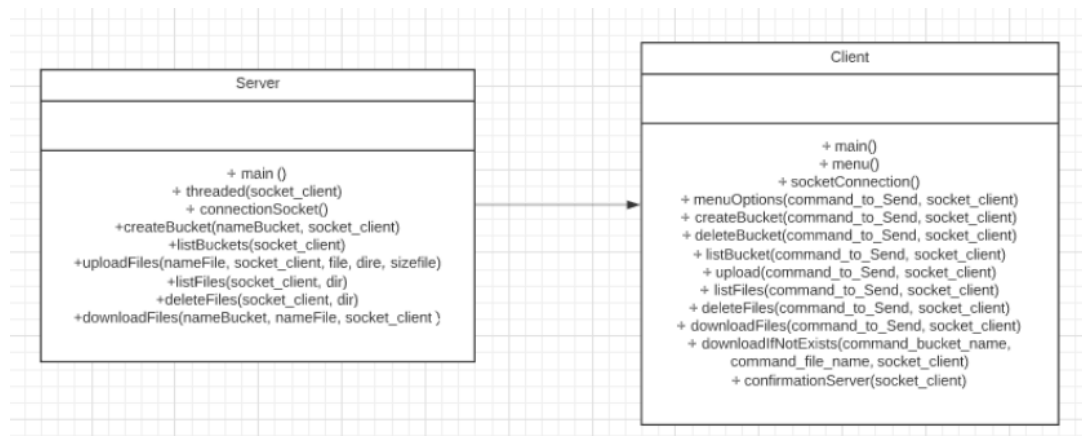
En este documento se plantean los principales objetivos y la metodología usada para lograr un protocolo de comunicación entre una aplicación cliente y servidor. Así mismo, se describe el servicio a implementar y el vocabulario de mensajes para su correcto funcionamiento.

OBJETIVOS

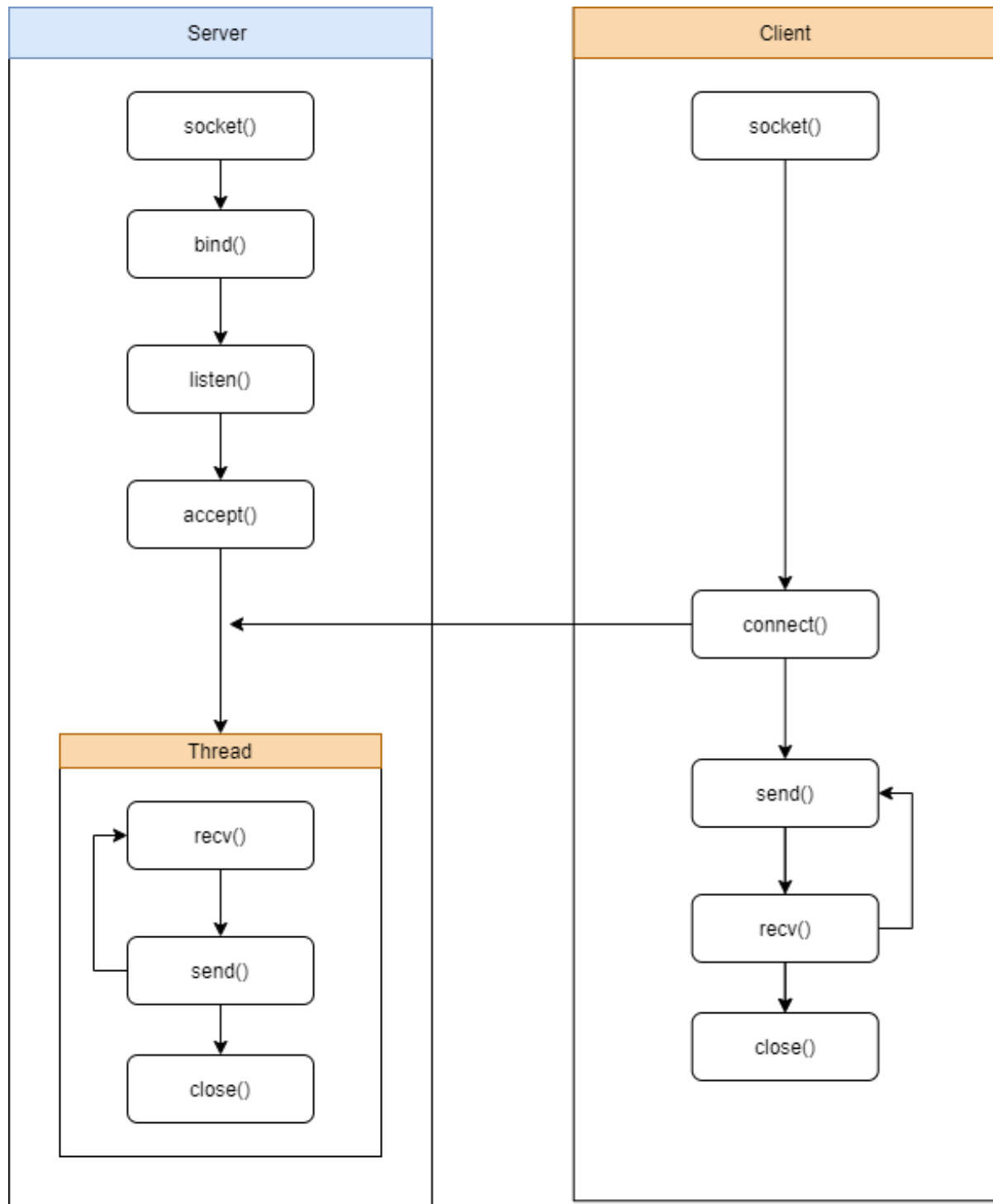
- Plantear de forma correcta el protocolo socket en el programa realizado y así poder tener la conexión entre máquinas.
- Relacionar los temas vistos en clase con los conocimientos ya adquiridos.
- Comprender cómo funciona la conexión socket relacionado con diferentes procesos de archivos como subir, descargar, eliminar etc...
- Desarrollar habilidades en los aspectos relacionados de la programación de aplicaciones en red.

ESPECIFICACIÓN FORMAL DEL PROTOCOLO

Diagrama de clases:



Descripción gráfica del proceso de conexión de los sockets:



DESCRIPCIÓN DEL SERVICIO

El servicio implementado se basa en un protocolo de comunicación entre una aplicación cliente y servidor, que tiene como fin el almacenamiento de datos, alojando diferentes tipos de archivos como: documentos, imágenes, archivos binarios, etc. Así pues, cada archivo se almacena en contenedores llamados buckets. Cabe destacar que estas carpetas se almacenan en la ruta donde se ejecuta la aplicación.

Este servicio de almacenamiento permite a los clientes:

- Crear buckets
- Eliminar buckets
- Listar buckets
- Subir archivos a un bucket
- Listar archivos de un bucket
- Eliminar archivos de un bucket
- Descargar archivos desde un bucket

Nota: Los buckets creados al ejecutar el programa se crearán dentro de una carpeta llamada *Buckets* que se creará en la ruta donde es ejecutado el protocolo. Así mismo, al descargar un archivo estos se almacenarán en una carpeta llamada *Downloads* que se creará en la ruta donde es ejecutado el protocolo.

VOCABULARIO DE MENSAJES

El conjunto de mensajes que implementa el protocolo está dado por el siguiente menú:

Option	Name Option
1	Create bucket
2	Delete bucket
3	List bucket
4	Upload bucket
5	List files
6	Dele files
7	Download files

Dentro del funcionamiento del protocolo se deben ingresar algunas de las opciones desplegadas en el menú, las cuales efectúan diferentes acciones dentro de la aplicación.

Cabe destacar que, para facilidad del usuario, al ingresar una opción se le notificarán por medio de mensajes los diferentes requisitos que debe ingresar para ejecutar el protocolo de manera correcta.

Opción 1:

Esta opción hace referencia a crear un bucket, como requisito se le debe ingresar el nombre respectivo del bucket.

Opción 2:

Esta opción hace referencia a eliminar un bucket, como requisito se le debe ingresar el nombre respectivo del bucket que desea borrar.

Opción 3:

Esta opción tiene como función listar todos los buckets creados por los clientes, no tiene ningún requisito para su funcionamiento.

Opción 4:

Tiene como función subir archivos almacenados dentro de la máquina del cliente a un bucket creado dentro del protocolo. Como requisitos se le solicita el nombre del bucket al que desea subir el archivo, el nombre del archivo con su respectiva extensión (Ej: nombre.png) y el path donde se encuentra el archivo en el cliente.

Nota: La ruta o path debe tener su estructura con / más no con \ , por ejemplo:

Manera incorrecta: C:\User\Desktop

Manera correcta: C:/User/Desktop

Opción 5:

Esta opción tiene como función listar todos los archivos almacenados dentro de un bucket específico. Como requisito para su funcionamiento se requiere ingresar el nombre del bucket al cual desea listar los archivos.

Opción 6:

Esta opción hace referencia a eliminar un archivo almacenado dentro de bucket, como requisito se le debe ingresar el nombre respectivo del bucket.

Opción 7:

Tiene como función descargar archivos almacenados dentro del servidor en un bucket en específico a una carpeta denominada downloads en la máquina del cliente. Como requisitos se le solicita el nombre del bucket donde se encuentra almacenado el archivo y el nombre del archivo con su respectiva extensión (Ej: nombre.png)

Nota: Al ejecutar cada comando e ingresar los requisitos que se piden, se preguntará si desea seguir haciendo otras funcionalidades del protocolo, el cliente debe responder con la letra y si desea continuar, de lo contrario al ingresar otra letra el cliente finalizará su ejecución.

REGLA DE PROCEDIMIENTOS

Este protocolo de comunicación envía y recibe mensajes por parte del cliente y del servidor, así pues, cada vez que se realiza alguna funcionalidad de manera correcta, se proporciona un error o se requiere que se ingresen los requisitos se envía mensaje al cliente y así mismo se notifica en el servidor.

Procesamiento de mensajes:

- **Conexión exitosa:** En el servidor se mostrará un mensaje con la dirección y el puerto del cliente que se ha conectado.
- **Comandos exitosos:** Cada vez que se realice una funcionalidad del protocolo de manera correcta se le notificará al cliente que se ha creado, eliminado o listado exitosamente.
- **Manejo de errores:** En este caso se visualizaron los posibles errores que podrían surgir al ser utilizados por un cliente, dentro de los cuáles se encuentran:

En opción 1: No ingresar el nombre de un bucket.

En opción 2: Ingresar el nombre de un bucket incorrecto o que no exista.

En opción 3: No se maneja ningún error.

En opción 4: Ingresar el nombre de un bucket incorrecto o que no exista, no ingresar el nombre de un archivo o un path, un documento que no exista en la dirección especificada, o que por otra parte ya exista el mismo documento en un bucket.

En opción 5: No exista el bucket ingresado en los requisitos.

En opción 6: Ingresar el nombre de un bucket incorrecto o que no exista.

En opción 7: Ingresar el nombre de un bucket incorrecto o que no exista, no poner el nombre de un archivo, que ya exista en la carpeta downloads o que no exista en el bucket especificado.

Para cada error se le envía un mensaje al cliente notificándole lo sucedido.

CONCLUSIONES

- El protocolo socket permite conectar dos procesos si se desea desde computadoras distintas los cuales intercambian flujo de datos de una forma fiable y ordenada.
- El manejo de librerías facilita la implementación de conexiones y funcionalidades que se dan respecto al servidor y al cliente.