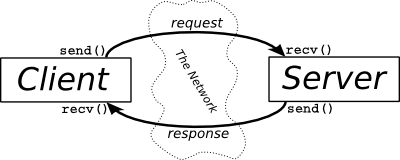
**WhatsUP**

**Aplicación, desarrollada con Java y JavaFX, de comunicación entre diversos usuarios.**

**En este documento se encontrarán los avances del proyecto “WhatsUP” donde se logró enviar mensajes de “Clientes” a “Cliente” a través del “Servidor”.**

Introducción:

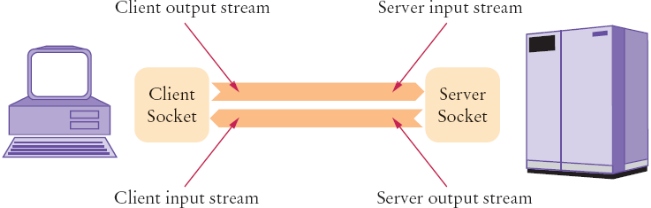
Para entender cómo funciona este programa es indispensable entender términos básicos de este tipo de sistemas. Principalmente conoceremos qué es un servidor, un cliente y un socket. Debemos entender algunos términos complementarios para saber cómo son ejecutados los programas de manera local (todo en la misma computadora).



Sockets:

En palabras sencillas, un socket es aquel proceso que mantiene la comunicación entre entidades (programas). Estos programas, o entidades, deben tener la capacidad de intercambiar entre sí cualquier secuencia de datos.

Dentro de este proyecto, los sockets son importantes porque logran mantener un sistema “cliente-servidor”. Esto nos permite lograr la comunicación entre nuestras dos entidades son el “cliente” y el “servidor”.



Servidor:

Sistema que contiene datos o proporciona recursos a los clientes que deben acceder. Normalmente son servidores de archivos que almacenan archivos, servidores de nombres que almacenan nombres y direcciones, etc. Cuando se habla de servidor es para denominar un sistema que proporciona servicios de inicio y sistemas de archivos a otros sistemas.

Archivos importantes dentro del proyecto:

* Server.java

Solamente ocupamos un archivo para generar el servidor y se compone de la siguiente manera:

1. Importar los paquetes: Aquí importamos todos lo paquetes y archivos que se vayan a utilizar para que este archivo se ejecute correctamente.
2. Clase “Server”: Se representa al servidor. Escucha las conexiones entrantes y crea un hilo “ClientHandler”, explicado en el siguiente punto, para cada cliente creado.
3. Clase “ClientHandler”: Representa el manejador de clientes. Cada instancia de esta clase maneja la comunicación con un cliente específico. Cada cliente tiene su propio hilo y administra el envío y la recepción de menajes.
4. Método “run” de “ClientHandler”: Maneja la recepción y procesamiento de menajes del cliente.

Cliente:

Es un sistema que solicita servicios o datos de un servidor. Un cliente puede solicitar código de programa actualizado o el suso de aplicaciones. Igualmente, se pueden solicitar archivos y datos para la entrada de datos.

Archivos importantes dentro del proyecto:

* App.java

Este archivo es el principal dentro de todo el programa.

1. Importar paquetes.
2. Clase “App”: Esta clase ayuda a construir la aplicación y que tengamos una interfaz gráfica previamente diseñada.
3. Método “start”: Este método se manda llamar cuándo la aplicación se inicia. Genera una conexión y despliega la ventana donde el usuario va a interactuar.
4. Método “setRoot”: Permite cambiar el contenido de la raíz de la escena. (En este avance solamente se maneja una escena).
5. Método “main”: Utiliza el método “launch” para iniciar la aplicación de JavaFX.

* Client.java

Aquí tenemos la mayor parte de la lógica que maneja el programa del cliente.

1. Definimos nuestra clase “Client”.
2. Aquí es importante definir el puerto que se va a utilizar la comunicación. Se guarda en la variable “ServerPort”
3. Igualmente, se defina la dirección IP como la misma de la computadora donde se están ejecutando los programas.
4. Hilo “sendMessage”: Esto nos ayuda a mandar mensajes de forma asíncrona. Ayuda a correr diversos procesos al mismo tiempo.
5. Hilo “readMessage”: Permite leer mensajes al mismo tiempo en que se mandan los mensajes. Ayuda a prevenir el choque de procesos.

* PrimaryController.java

Este archivo ayuda a leer todo lo que se introduce y ejecuta en la interfaz gráfica. Ya sea introducir texto, mostrar texto o presionar el botón de enviar.

1. Tenemos un “TextArea” que ayuda a mostrar los mensajes que son recibidos.
2. Un “TextField” es utilizado para que el usuario pueda introducir texto y lo mande a otro usuario.
3. El “Button” permite mandar el texto que el usuario desea compartir con otro usuario una vez que termina de escribir toda la cadena de texto.

* Primary.fxml

Este archivo contiene el diseño de la interfaz gráfica que el usuario va a utilizar. Se utilizó SceneBuilder. Esta herramienta ayuda a un diseño más sencillo. Es como si utilizaras Figma o Adobe XD.

Bibliografía:

* IBM documentation. (s. f.). <https://www.ibm.com/docs/es/aix/7.1?topic=systems-client-server>
* ¿Qué son los servidores, los clientes y los dispositivos? - Guía de Administración del Sistema: Administración básica. (2011, 1 enero). <https://docs.oracle.com/cd/E24842_01/html/E23289/clientconcept-33212.html>
* Team, K. (2023, 16 mayo). ¿Qué es un socket? | KeepCoding Bootcamps. KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-un-socket/>