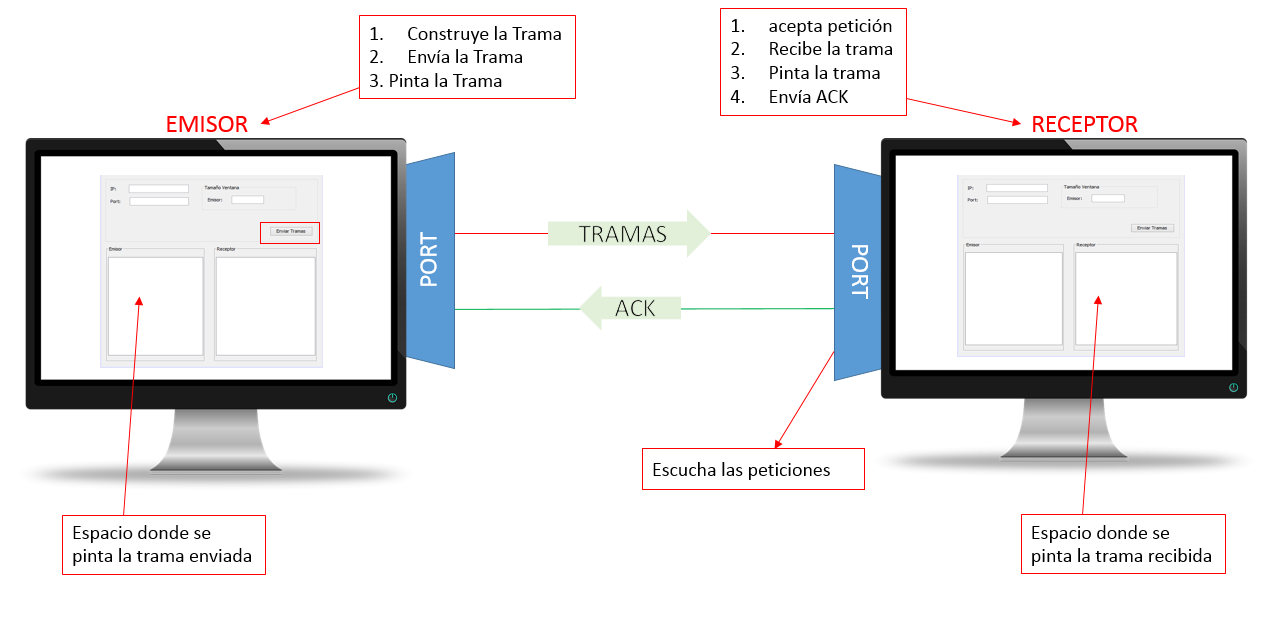
**VENTANA DESLIZANTE**

**Por:** CARLOS EDUARDO BARRERA

**EXPLICACION GENERAL:**

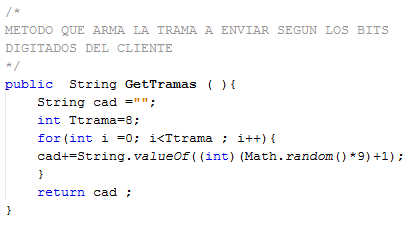
Este algoritmo funciona por el principio de sockets con cliente/servidor, en este caso sería emisor/receptor, es básicamente el funcionamiento de un chat (intercambio de mensajes) pero a bajo nivel, con el uso de la creación de tramas que es el contenido a enviar (emisor) y el contenido que se recibe (receptor).

Se utilizó el protocolo de ventana deslizante, el cual como se mencionó anteriormente, permite el intercambio de datos entre un emisor y un receptor por medio del manejo de tramas en segmentos (completas en el caso de esta solución y en ambos sentidos, o sea, dúplex), de las cuales el receptor confirma la llegada de estas y valida que hayan llegado correctamente, esto se hace de manera bidireccional (lo cual se explicará más adelante en el código).



**EXPLICACIÓN DE LA SOLUCIÓN (CÓDIGO):**

Para la implementación de la solución se utilizó el entorno del IDE de Netbeans, se utilizaron 2 clases: negocio, la cual contiene los métodos de la lógica del negocio: Emisor y la creación de tramas (getTramas)

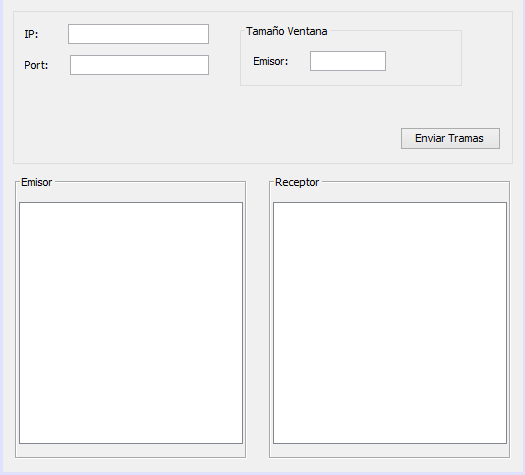


Este método crea la trama con un número aleatorio de 0 a 9, con una cantidad de 8 cifras por medio de un bucle y finalmente retorna la cadena a mostrar.

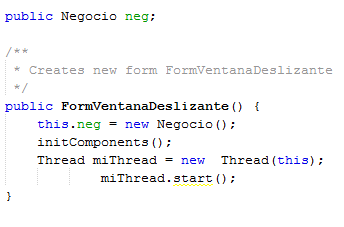


El método del emisor recibe la IP del host al cual se le va a enviar las tramas, recibe la trama a enviar y el número del puerto por el cual el receptor va a recibir, crea un dato booleano que servirá como bandera, y un Socket que será la conexión de emisor/receptor, se crea el socket con la dirección IP y el número del puerto, además de esto lee que envía el servidor con un buffer (ACK acuse de recibo), después de enviar la petición y mostrar la respuesta en el receptor cierra el socket.

Para la clase del formulario de la ventana deslizante que es el Frame de la GUI se ve de la siguiente manera:

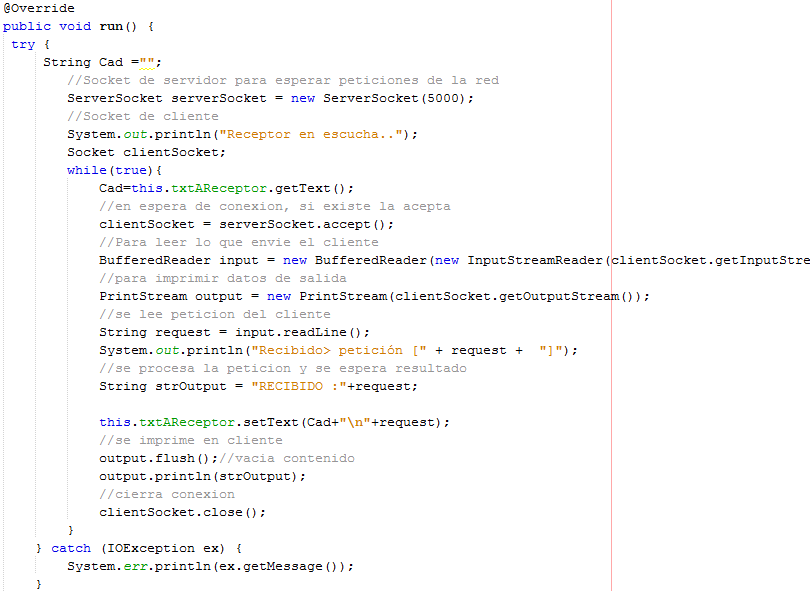


Para el funcionamiento de esto, creamos una instancia de la clase negocio, y en donde se inicializan los componentes, se creó un hilo que será el hilo de “escucha” del servidor.



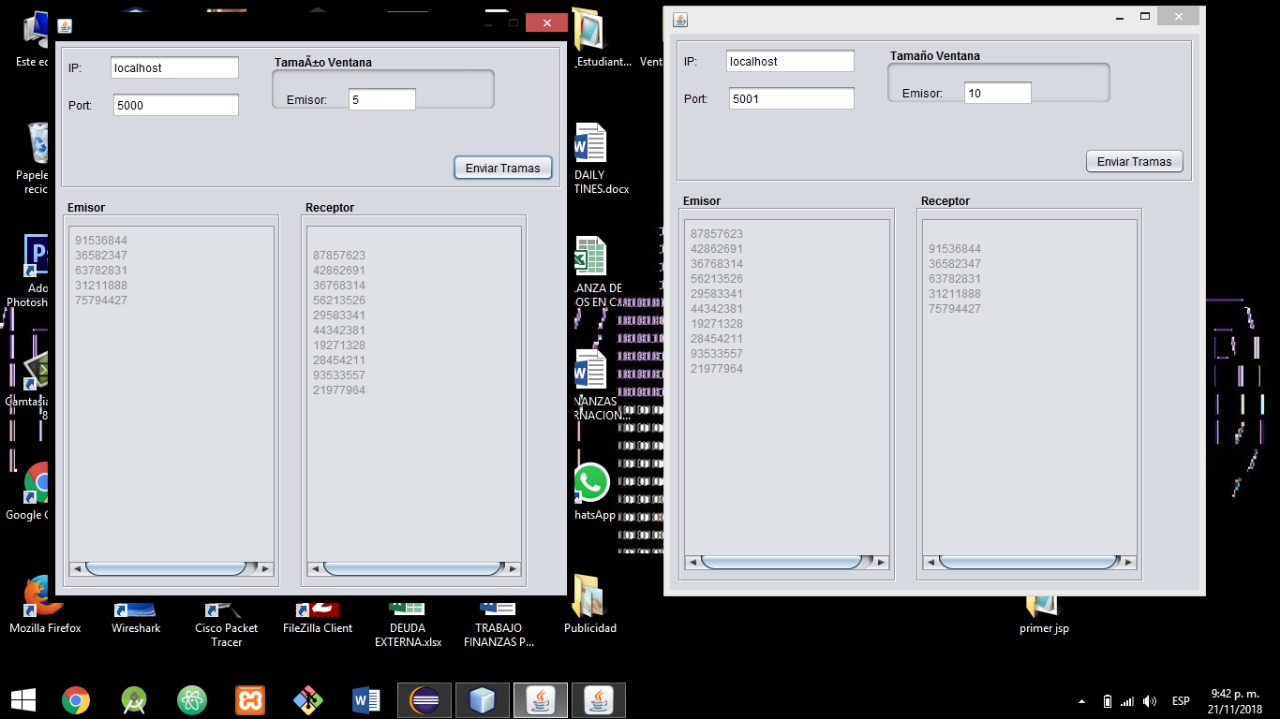


Para el funcionamiento del botón “Enviar tramas” que se encuentra en la GUI, llenamos los campos de texto, donde tenemos, la IP, el puerto, y el “EMISOR” es la cantidad de tramas que se van a enviar de una ventana a otra, los cuales se mostraran en el campo “emisor” y en la otra ventana (diferente puerto, e IP) en la ventana “receptor” recibirá el contenido enviado y confirmado (Que será la cantidad de tramas enviadas por el emisor).



El receptor utiliza un hilo para estar en continua escucha por medio de un puerto en específico. En este método creamos el ServerSocket y le asignamos un puerto de escucha, instanciamos el clientSocket e hicimos que este acepte todas las peticiones entrantes. Luego de esto Leemos lo que nos envía el emisor enviamos el ACK (acuse de recibo) y pintamos en el área de receptor.

Cuando ponemos a funcionar el programa desde dos computadores en un área local, o como para cuestiones de ejemplo se usaron dos IDE distintas para ejecutar el código desde dos instancias diferentes: Netbeans y Eclipse.



Se usó localhost como IP porque para cuestiones de ejemplo como se mencionó anteriormente, se hizo desde dos IDE distintas, el puerto se cambió para su funcionamiento: 5000 y 5001, y como se muestra en la imagen, debido a la cantidad de tramas, que se escribieron en el cuadro, se enviaron y se recibieron de manera correcta, el espacio del receptor no tiene un límite definido, por eso recibe cualesquiera cantidad de tramas que el otro emisor envié.