

6/2/26



Proyecto Segundo parcial

Materia:

APLICACIONES DISTRIBUIDAS

Integrantes:

Yagual Alarcón Mariana

Sánchez Marcillo Angelo

Basurto Delgado Iván

Profesor:

Ing. Joel Barba Salazar

Curso:

SOF-S-NO-9-5

Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

Índice

PROYECTO SEGUNDO PARCIAL -SISTEMA DE CHAT DISTRIBUIDO	2
1. INTRODUCCIÓN	2
2. DISEÑO Y ARQUITECTURA DEL SISTEMA	2
2.1 Descripción general del sistema	2
2.2 Arquitectura cliente–servidor	3
Esquema de la arquitectura:	3
2.3 Tecnologías utilizadas	3
3. IMPLEMENTACIÓN DEL SERVIDOR	4
3.1 Gestión de conexiones concurrentes	4
3.2 Autenticación y gestión de usuarios	5
3.3 Seguridad e integridad de los datos	5
4. IMPLEMENTACIÓN DEL CLIENTE	6
4.1 Interfaz gráfica del cliente	6
4.2 Comunicación en tiempo real	7
5. PRUEBAS REALIZADAS	7
6. DOCUMENTACIÓN Y PRESENTACIÓN	8
7. CONCLUSIONES	9

Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

PROYECTO SEGUNDO PARCIAL -SISTEMA DE CHAT DISTRIBUIDO

1. INTRODUCCIÓN

El presente proyecto aborda el desarrollo de un Sistema de Chat Distribuido basado en una arquitectura cliente–servidor, diseñado para permitir la comunicación en tiempo real entre múltiples usuarios conectados simultáneamente. Este sistema fue desarrollado como parte de la asignatura Aplicaciones Distribuidas, con el propósito de aplicar de manera práctica los conceptos teóricos relacionados con la comunicación en red, la concurrencia y la interacción entre procesos distribuidos.

El sistema permite a los usuarios establecer una conexión con un servidor central, autenticarse mediante un nombre de usuario y participar en una conversación compartida. Los mensajes enviados por cada usuario son recibidos por el servidor y redistribuidos de forma inmediata a todos los clientes conectados, garantizando una comunicación eficiente, ordenada y confiable.

2. DISEÑO Y ARQUITECTURA DEL SISTEMA

2.1 Descripción general del sistema

El sistema desarrollado está compuesto por dos componentes principales: el servidor y el cliente. El servidor actúa como el núcleo central de la aplicación y es responsable de aceptar las conexiones entrantes, gestionar los usuarios conectados, recibir los mensajes enviados por los clientes y reenviarlos a todos los usuarios activos. Por su parte, el cliente proporciona una interfaz gráfica que permite a los usuarios interactuar con el sistema de chat de manera intuitiva y sencilla.

Esta separación de responsabilidades permite una organización clara del sistema, facilitando su mantenimiento, escalabilidad y comprensión, aspectos fundamentales en el desarrollo de aplicaciones distribuidas.

Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

2.2 Arquitectura cliente–servidor

La arquitectura del sistema se basa en el modelo cliente–servidor, donde múltiples clientes se conectan a un servidor central mediante el uso de sockets TCP. Cada cliente se ejecuta como un proceso independiente y mantiene una conexión persistente con el servidor durante toda la sesión de comunicación.

El servidor implementa un modelo concurrente utilizando hilos, de tal manera que cada cliente conectado es atendido por un hilo independiente. Este enfoque permite que varios usuarios puedan enviar y recibir mensajes simultáneamente sin que la comunicación de uno interfiera con la de los demás, garantizando así un funcionamiento fluido y eficiente del sistema.

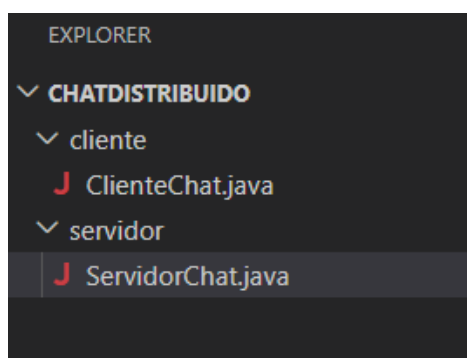
Esquema de la arquitectura:

Cliente 1

Cliente 2 → Servidor de Chat Distribuido ← Cliente 3

Cliente n

Diagrama o estructura del proyecto



2.3 Tecnologías utilizadas

Para el desarrollo del sistema se emplearon tecnologías ampliamente utilizadas en el ámbito del desarrollo de software distribuido. El lenguaje de programación Java fue seleccionado debido a su portabilidad, robustez y amplio soporte para programación en red.

Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

y concurrencia. La comunicación entre el cliente y el servidor se realiza mediante sockets TCP, los cuales garantizan la integridad y fiabilidad de los datos transmitidos.

Asimismo, se utilizó la biblioteca Java Swing para la implementación de las interfaces gráficas, permitiendo crear ventanas, botones y componentes visuales de manera eficiente. El uso de hilos (Threads) permitió gestionar múltiples conexiones simultáneamente, asegurando un correcto funcionamiento del sistema bajo escenarios con varios usuarios conectados.

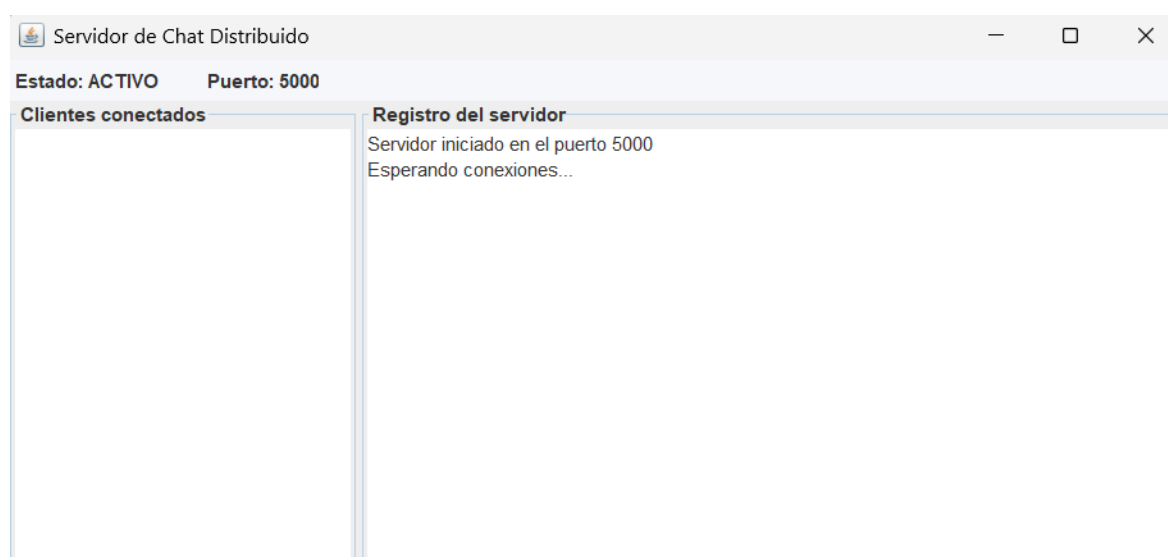
3. IMPLEMENTACIÓN DEL SERVIDOR

3.1 Gestión de conexiones concurrentes

El servidor fue implementado utilizando la clase `ServerSocket`, la cual permite escuchar solicitudes de conexión entrantes en un puerto específico. Cuando un cliente solicita una conexión, el servidor la acepta y crea un hilo independiente encargado de gestionar la comunicación con dicho cliente.

Este mecanismo de concurrencia es fundamental en aplicaciones distribuidas, ya que permite que múltiples clientes interactúen con el servidor de forma simultánea sin generar bloqueos ni interrupciones en el servicio.

Servidor en ejecución



Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

3.2 Autenticación y gestión de usuarios

Como mecanismo de autenticación básica, cada cliente debe ingresar un nombre de usuario al iniciar la aplicación. Este nombre es enviado al servidor al momento de establecer la conexión y se utiliza como identificador único dentro del sistema de chat.

El servidor mantiene una lista actualizada de los usuarios conectados, permitiendo gestionar correctamente las conexiones y desconexiones. Cada vez que un usuario se conecta o se desconecta, el servidor actualiza esta lista y notifica a todos los clientes, asegurando que la información mostrada sea consistente en tiempo real.

Lista de usuarios conectados en el servidor



3.3 Seguridad e integridad de los datos

La comunicación entre el servidor y los clientes se realiza utilizando el protocolo TCP, el cual proporciona mecanismos de control que garantizan la integridad, confiabilidad y entrega ordenada de los datos transmitidos. Esto asegura que los mensajes enviados por los usuarios lleguen completos y en el orden correcto, evitando pérdidas o duplicaciones durante la comunicación.

Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

Si bien el sistema no implementa cifrado avanzado, el uso de TCP resulta adecuado para el contexto académico del proyecto, cumpliendo con los requisitos de integridad y seguridad establecidos.

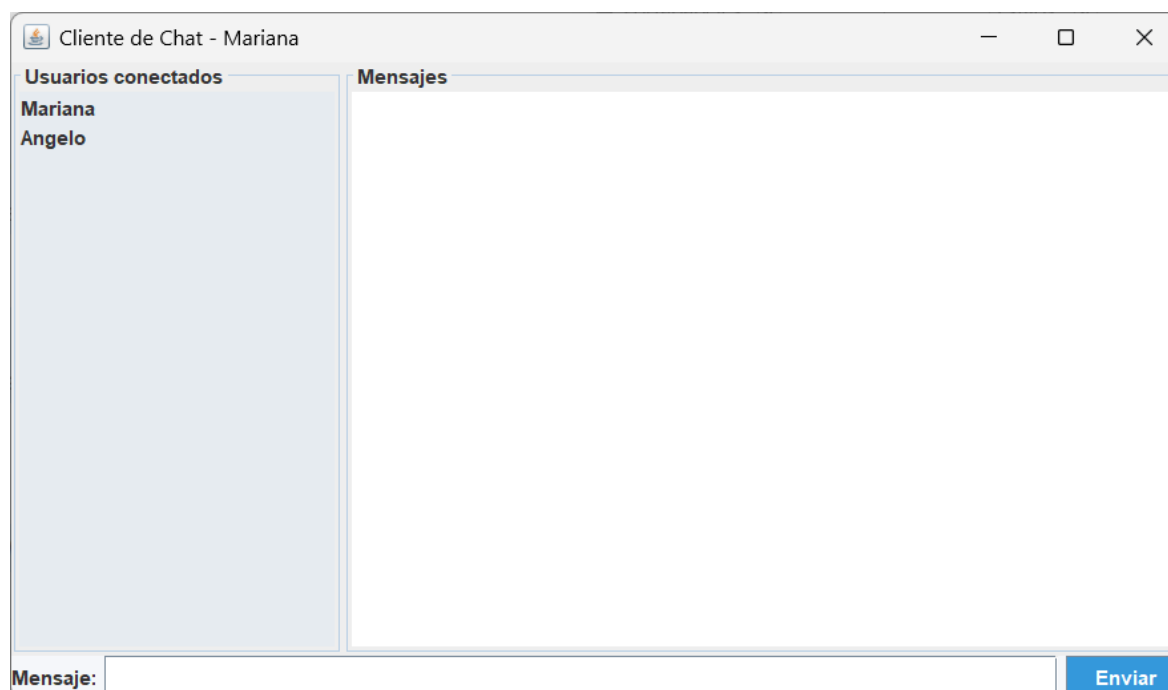
4. IMPLEMENTACIÓN DEL CLIENTE

4.1 Interfaz gráfica del cliente

El cliente cuenta con una interfaz gráfica desarrollada en Java Swing, diseñada siguiendo criterios de usabilidad, claridad visual y simplicidad. La interfaz permite al usuario ingresar su nombre de usuario, enviar mensajes, recibir mensajes en tiempo real y visualizar la lista de usuarios conectados.

La disposición de los componentes y el uso de una paleta de colores suaves contribuyen a una experiencia de usuario agradable y profesional, facilitando la interacción con el sistema de chat.

Interfaz gráfica del cliente



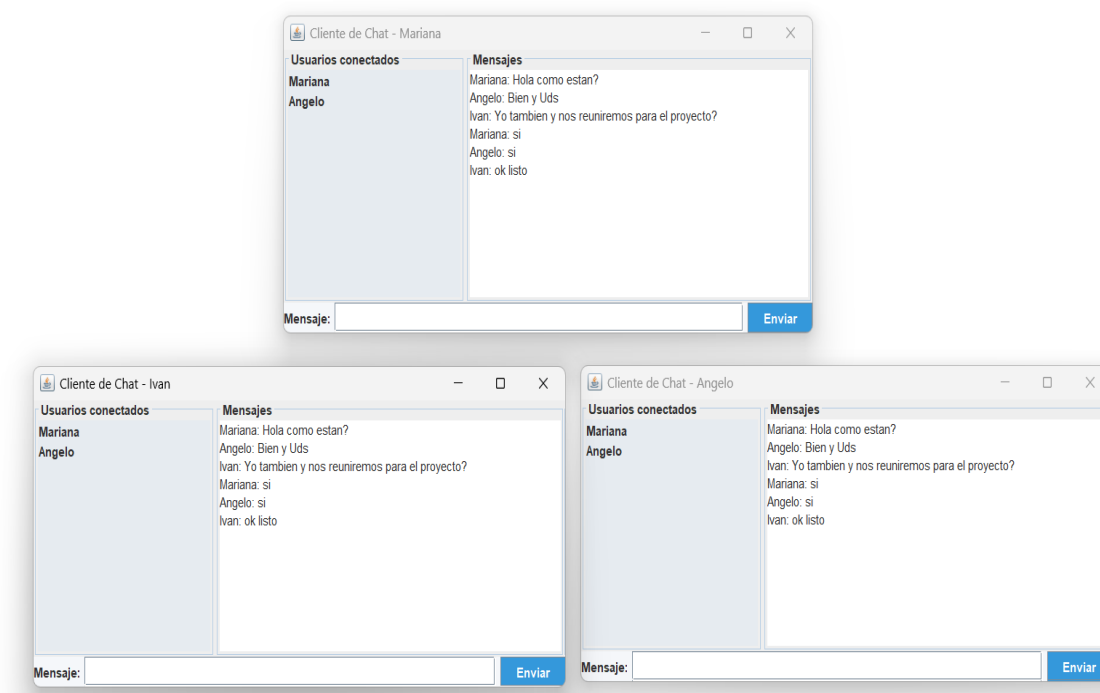
Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

4.2 Comunicación en tiempo real

El cliente establece una conexión persistente con el servidor mediante sockets TCP. Para evitar bloqueos en la interfaz gráfica durante la recepción de mensajes, se implementa un hilo independiente encargado de escuchar los mensajes enviados por el servidor.

Este enfoque garantiza que la comunicación se realice en tiempo real, permitiendo al usuario enviar y recibir mensajes de manera fluida y sin interrupciones.

Comunicación en tiempo real



5. PRUEBAS REALIZADAS

Las pruebas del sistema se llevaron a cabo ejecutando el servidor y múltiples instancias del cliente en una sola computadora, simulando un entorno distribuido. Cada instancia del cliente se ejecuta como un proceso independiente, lo que permite evaluar el comportamiento del sistema ante múltiples conexiones simultáneas.

Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

Durante las pruebas se verificó el correcto funcionamiento del sistema en términos de conexión de clientes, envío y recepción de mensajes, actualización de la lista de usuarios y estabilidad general de la aplicación.

Múltiples clientes conectados



6. DOCUMENTACIÓN Y PRESENTACIÓN

El proyecto cuenta con una documentación técnica clara y detallada que describe la arquitectura del sistema, la implementación del servidor y del cliente, así como los pasos necesarios para su correcta ejecución dentro del entorno de desarrollo Visual Studio Code. Para la ejecución del sistema, primero se inicia el archivo ServidorChat.java mediante clic derecho sobre el archivo y seleccionando la opción “Run Java”, lo que permite levantar el servidor y dejarlo en espera de conexiones.

Posteriormente, se ejecuta el archivo ClienteChat.java de la misma manera, pudiendo abrir varias instancias del cliente para simular múltiples usuarios conectados al servidor. El código fuente se encuentra organizado en paquetes diferenciados para servidor y cliente, y debidamente comentado, lo que facilita su comprensión, mantenimiento y futura ampliación.

Asignatura	Datos del alumno	Fecha
Aplicaciones Distribuidas	Profesor: Ing. Joel Barba Salazar	06-02-2026
	Paralelo: SOF-S-NO-9-5	

7. CONCLUSIONES

El desarrollo del Sistema de Chat Distribuido permitió aplicar de manera práctica los conceptos fundamentales de las aplicaciones distribuidas, tales como la arquitectura cliente–servidor, la comunicación en red mediante sockets y la concurrencia a través del uso de hilos. Durante la implementación se evidenció la importancia de una correcta gestión de múltiples conexiones simultáneas, así como de la integridad y el orden de los mensajes intercambiados entre los diferentes clientes conectados al sistema.

El sistema desarrollado cumple con los requisitos establecidos, permitiendo la comunicación en tiempo real entre múltiples usuarios, la visualización dinámica de la lista de clientes conectados y una interacción clara a través de interfaces gráficas desarrolladas en Java Swing. En conclusión, el proyecto constituye una solución funcional y eficiente para un entorno distribuido a nivel académico, y representa una base sólida para futuras mejoras, como la incorporación de mecanismos de seguridad más avanzados o su despliegue en entornos reales.