

Chat Grupal

Programación de Servicios y Procesos

Cristian Fernández

21 de enero de 2026

Índice

1. Descripción del problema	2
2. Requisitos funcionales	3
3. Requisitos no funcionales	4
4. Casos de uso	5
5. Historias de usuario	6
6. Herramientas de desarrollo	7
7. Lenguajes de programación	8
8. Interfaz de la aplicación	9

1. Descripción del problema

El docente encargado de impartir el módulo de *Programación de Servicios y Procesos*, Don Roberto Castro Liste, ha encargado la realización de una tarea que consiste en crear un aplicación en **JavaFX** que permita chatear entre usuarios, demostrando la comprensión y el dominio en el uso de sockets y también en el uso de hilos con la utilización de métodos como *newFixedThreadPool()* de la clase *ExecutorService*.

El programa debe seguir estas pautas:

- Crear una clase servidor que esté escuchando constantemente conexiones entrantes.
- Crear una clase cliente que permita enviar y recibir mensajes del servidor.
- Crear una clase manejadora que comunique correctamente las 2 clases anteriores.
- Diseñar una interfaz que permita ver de forma clara el funcionamiento del chat.

2. Requisitos funcionales

A continuación se listan los requisitos funcionales del sistema:

RF-1: Solicitar al usuario que introduzca su nombre antes de entrar en el chat.

RF-2: Pulsar un botón que inicie la ejecución del programa en una nueva ventana.

RF-3: Permitir la inserción de mensajes de forma ágil únicamente usando el teclado.

RF-4: Mostrar correctamente y en tiempo real todos los mensajes de todos los participantes en el chat.

RF-5: Terminar los hilos en ejecución cuando se cierre la aplicación Cliente.

3. Requisitos no funcionales

A continuación se listan los requisitos NO funcionales del sistema:

RNF-1: El tiempo de respuesta al pulsar el botón *Entrar Chat* no debe superar los 2 segundos.

RNF-2: La interfaz gráfica de usuario ha de ser fácilmente entendible.

RNF-3: El tiempo de respuesta al mandar un mensaje pulsando Enter no debe superar el segundo.

RNF-4: El usuario debe distinguir fácilmente sus mensajes del resto de mensajes de los demás usuarios.

4. Casos de uso

A continuación se muestran los casos de uso mediante la siguiente tabla:

Caso de uso	Descripción
Introducir nombre	El usuario debe introducir su nombre antes de entrar en el chat.
Iniciar aplicación	El usuario puede iniciar la ventana de chat pulsando un botón.
Escribir mensaje	El usuario puede enviar un mensaje escribiendo y pulsando Enter.
Cerrar aplicación	El usuario puede cerrar la aplicación por completo pulsando el botón de cerrar ventana.

5. Historias de usuario

A continuación se muestran las historias de usuario mediante la siguiente tabla:

ID	Como...	Quiero...	Para...
HU-01	Usuario	Insertar mi nombre	Que me identifiquen en el chat
HU-02	Usuario	Pulsar un botón	Iniciar la aplicación
HU-03	Usuario	Manejar el chat únicamente con el teclado	Disfrutar de una buena usabilidad
HU-04	Usuario	Ver los nombres de los participantes	Saber en todo momento con quién hablo

6. Herramientas de desarrollo

IntelliJ Community Edition es un IDE gratuito y de código abierto, ideal para el desarrollo en lenguajes JVM y Android. Sus características principales incluyen un potente editor de código con autocompletado inteligente y refactorización, soporte para control de versiones, un depurador integrado, y herramientas de pruebas unitarias. Está disponible para todas las plataformas.

Características principales:

- **Editor de código potente:** Ofrece resaltado de sintaxis, análisis en tiempo real, sugerencias de código, e inspecciones y correcciones rápidas.
- **Soporte para múltiples lenguajes:** Es compatible con Java, Kotlin, Groovy, Scala, y otros lenguajes JVM.
- **Integración con control de versiones:** Permite la integración con sistemas como Git, SVN, y Mercurial.

7. Lenguajes de programación

JavaFX se caracteriza por su capacidad de crear aplicaciones enriquecidas visualmente, integrando multimedia (audio, video) y gráficos vectoriales. Permite desarrollar aplicaciones multiplataforma (escritorio, web, móvil, smart TV). Otras características clave son la integración fluida con el ecosistema Java y herramientas como FXML y Scene Builder para un desarrollo más eficiente y colaborativo.

Características principales:

- **Interfaces gráficas de usuario (GUI) enriquecidas:** Permite crear interfaces modernas y atractivas con animaciones, gráficos vectoriales, y elementos personalizables.
- **Integración multimedia:** Facilita la inclusión de contenido de audio, video y otros medios dentro de las aplicaciones.
- **Desarrollo declarativo (FXML):** Utiliza FXML, un lenguaje basado en XML, para describir la interfaz de usuario, lo que facilita el trabajo colaborativo entre diseñadores y desarrolladores.

8. Interfaz de la aplicación

A continuación unas capturas de las diferentes vistas de la interfaz.

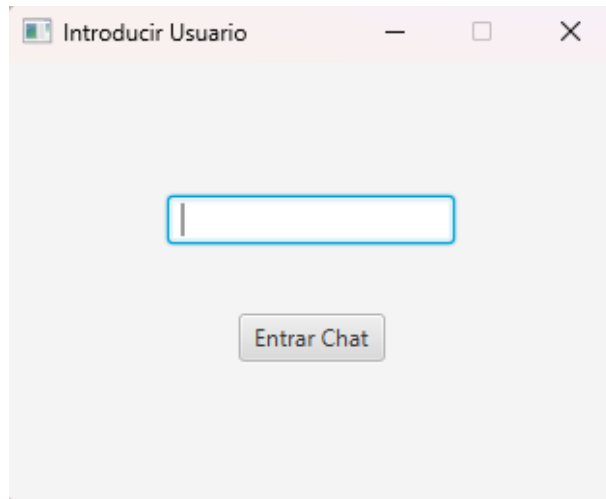


Figura 1: Vista inicial, requiere introducir nombre usuario para continuar



Figura 2: Si el campo está vacío se muestra un placeholder en el interior del TextField

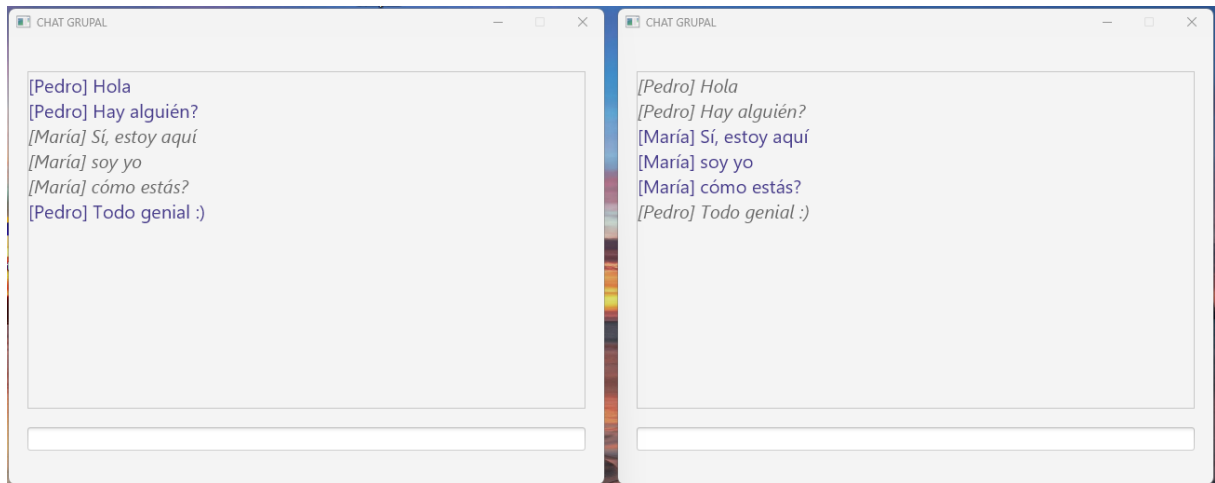


Figura 3: Los textos en color son los del usuario, en cursiva los de los demás