

PROGRAMACION DE SERVICIOS

Chat Grupal

Estudiante: John Arenales Fernandez

Profesor: Roberto Castro Liste

Año: 2026

Índice

1. Introducción	2
2. Descripcion del problema	3
3. Requisitos	4
3.1. Requisitos funcionales	4
3.2. Requisitos no funcionales	4
4. Casos de uso	5
5. Stack Tecnologico	6
5.1. Github	6
5.2. Figma	6
5.3. Java	6
5.4. JavaFx	6
5.5. IntelliJ IDEA	6
5.6. Visual Studio Code	6

1. Introducción

En la programación de servicios moderna, la comunicación entre aplicaciones es fundamental. Este proyecto implementa un **chat grupal en tiempo real** utilizando sockets y buffers, conceptos clave para entender cómo los sistemas se comunican a través de redes.

Un **socket** es un punto final de comunicación que actúa como intermediario entre dos programas que desean intercambiar información. Específicamente, utilizamos sockets TCP (Transmission Control Protocol), que garantizan una conexión confiable y ordenada entre el servidor y los clientes. Estos sockets utilizan **buffers** para almacenar temporalmente los datos antes de ser procesados, permitiendo que la información fluya eficientemente sin pérdida.

En esta práctica, implementamos una arquitectura cliente-servidor donde:

- El **servidor** mantiene múltiples conexiones simultáneas con diferentes clientes
- Cada **cliente** se conecta al servidor a través de un socket dedicado
- Los **buffers** almacenan los mensajes entrantes y salientes
- Los mensajes se distribuyen a todos los clientes conectados (broadcast)

Este proyecto nos permite aplicar y consolidar los conceptos fundamentales vistos en clase sobre:

- Comunicación por sockets (TCP/IP)
- Programación concurrente (manejo de múltiples hilos)
- Operaciones de entrada/salida en redes
- Sincronización entre procesos
- Desarrollo de aplicaciones cliente-servidor

2. Descripcion del problema

Como desarrolladores independientes, hemos sido contactados por una empresa del sector de las redes sociales tras identificar una limitación crítica en su plataforma: la falta de una interacción fluida entre sus usuarios. Actualmente, la aplicación carece de un servicio de mensajería interna, lo que impide la comunicación directa, ya sea de forma privada o grupal. El objetivo del proyecto es diseñar e implementar un sistema de mensajería instantánea integrado, que garantice la plena identificación de los interlocutores y permita a los usuarios personalizar su identidad mediante nombres de usuario de su elección.

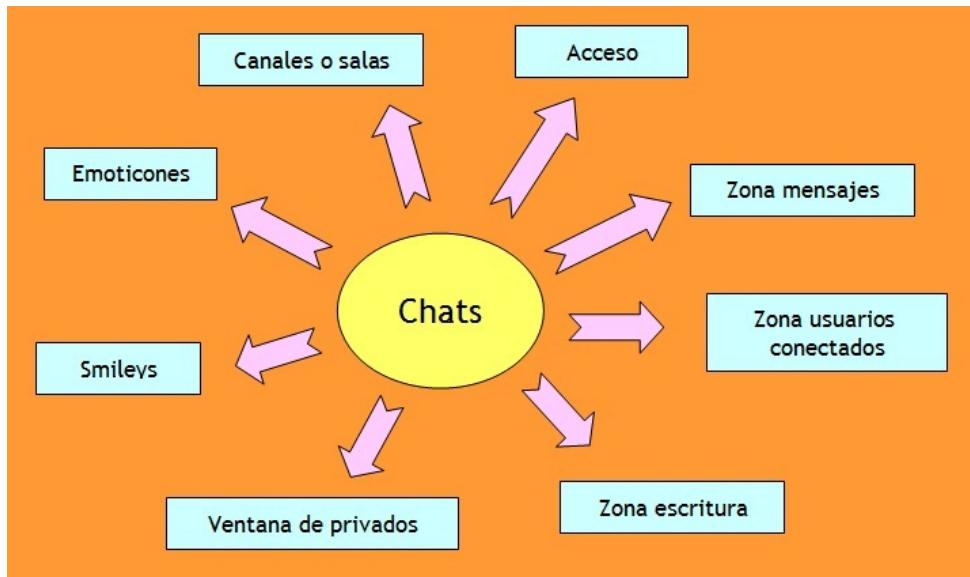


Figura 1: Componentes de un chat

3. Requisitos

Requisitos funcionales:

Definen las acciones y servicios que el sistema debe proporcionar. Especifican qué hace la aplicación, como las funcionalidades disponibles para el usuario, los procesos que realiza y la forma en que responde a determinadas acciones.

Requisitos no funcionales:

Describen las cualidades y restricciones del sistema. Indican cómo debe funcionar la aplicación, incluyendo aspectos como el rendimiento, la seguridad, la usabilidad, la compatibilidad o la fiabilidad.

3.1. Requisitos funcionales

- Permitir a los usuarios registrarse con un nombre de usuario único.
- Facilitar el inicio de sesión para usuarios registrados.
- Opción de restablecer la contraseña a través del correo electrónico.
- Interfaz para enviar y recibir mensajes en tiempo real.
- Capacidad de crear y gestionar grupos de chat.
- Funcionalidad para enviar archivos y imágenes.
- Notificaciones de mensajes nuevos.
- Cierre de sesión seguro.

3.2. Requisitos no funcionales

- Seguridad: Los datos del usuario y los mensajes deben estar cifrados.
- Escalabilidad: El sistema debe manejar un aumento en el número de usuarios sin degradar el rendimiento.
- Disponibilidad: El servicio debe estar operativo 24/7 con un tiempo de inactividad mínimo.
- Rendimiento: Los mensajes deben ser entregados en menos de un segundo.
- Usabilidad: La interfaz debe ser intuitiva y fácil de usar.

4. Casos de uso

- **Registro de usuario:** Un nuevo usuario se registra proporcionando un nombre de usuario y una contraseña.
- **Inicio de sesión:** Un usuario existente ingresa con sus credenciales para acceder a su cuenta.
- **Restablecimiento de contraseña:** Un usuario que olvidó su contraseña solicita un restablecimiento a través de su correo electrónico.
- **Envío de mensajes:** Un usuario envía un mensaje a otro usuario o a un grupo.
- **Recepción de mensajes:** Un usuario recibe un mensaje de otro usuario o de un grupo.
- **Creación de grupos:** Un usuario crea un nuevo grupo de chat e invita a otros usuarios a unirse.
- **Gestión de perfil:** Un usuario actualiza su información personal o cambia su contraseña.
- **Cierre de sesión:** Un usuario cierra sesión de su cuenta de forma segura.

5. Stack Tecnologico

5.1. Github

Plataforma para el control de versiones del proyecto, permitiendo un seguimiento detallado de los cambios y la colaboración entre desarrolladores.

5.2. Figma

Herramienta de diseño colaborativo utilizada para crear prototipos de la interfaz de usuario y definir la experiencia visual de la aplicación.

5.3. Java

Lenguaje de programación principal utilizado en el desarrollo del backend de la aplicación, garantizando robustez y escalabilidad.

5.4. JavaFx

Framework utilizado para construir la interfaz gráfica de usuario (GUI) en aplicaciones de escritorio, proporcionando una experiencia de usuario rica y interactiva.

5.5. IntelliJ IDEA

Entorno de desarrollo integrado (IDE) utilizado para escribir, depurar y ejecutar el código fuente en Java, mejorando la productividad del desarrollador.

5.6. Visual Studio Code

Editor de código fuente utilizado para editar archivos de configuración y otros lenguajes de programación, ofreciendo flexibilidad y ligereza en el desarrollo.