
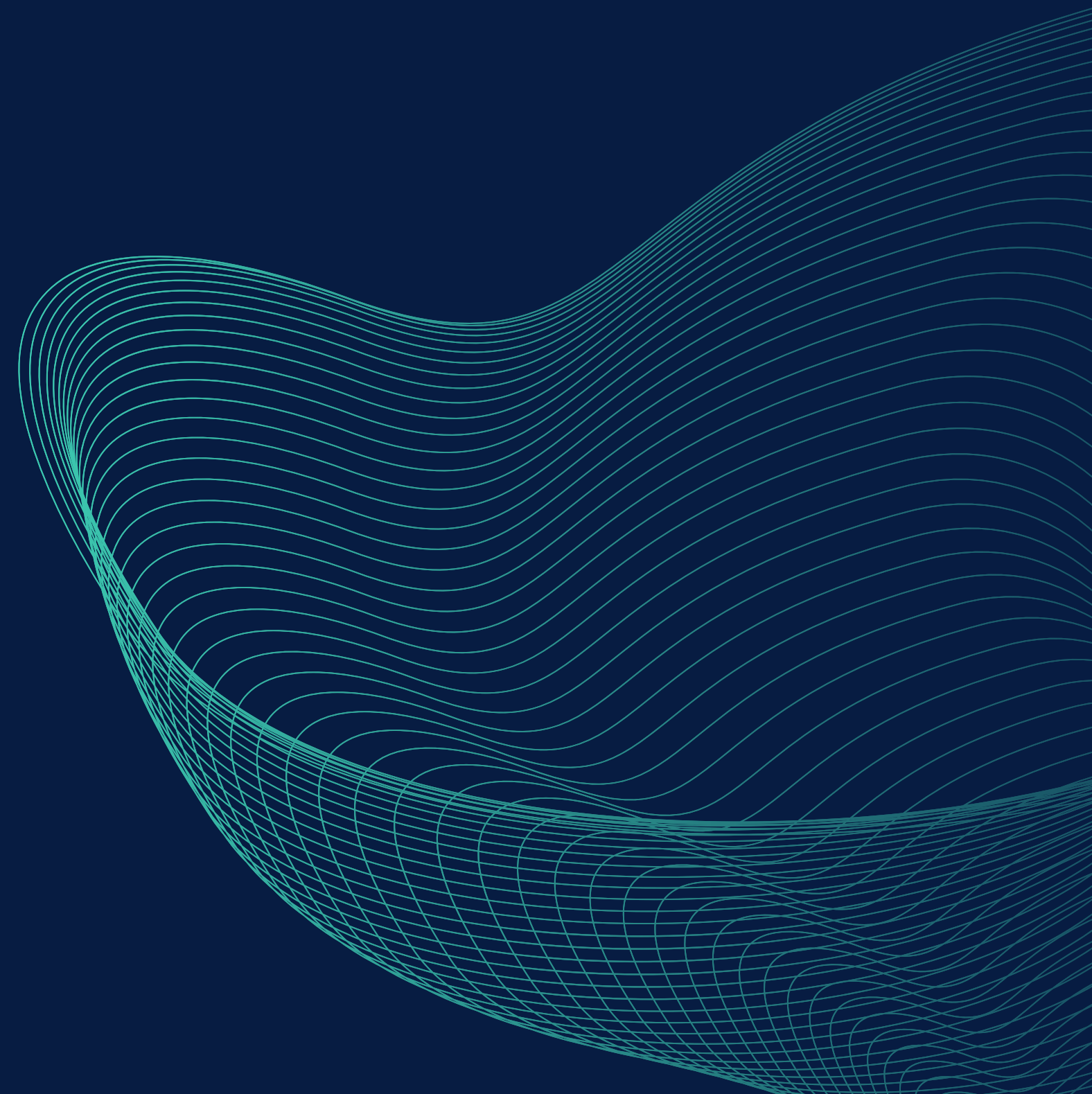




Proyecto chat XMPP

Daniel González 20293



Características implementadas

- Creación de cuenta: Desde la aplicación es posible crear una cuenta en el servidor, siempre y cuando no exista el nombre de usuario previamente. Luego de crearla se inicia sesión automáticamente.
- Inicio de sesión: Si ya se tienen credenciales en el servidor, es posible iniciar sesión en él.
- Eliminación de cuenta: Si se ha iniciado sesión en la aplicación, es posible eliminar la cuenta desde el proyecto.
- Cierre de sesión: Luego de iniciar sesión, es posible cerrarla. Con esta acción el usuario aparecera inactivo para los demás.
- Mostrar contactos y estado: Es posible mostrar los contactos que se tienen, junto con su estado de actividad, su mensaje personalizado y su disponibilidad.
- Agregar usuario a contactos: Brindando el nombre de usuario, se puede agregar este a los contactos. Además, todas las solicitudes de amistad de otros usuarios se aceptan automáticamente.

- Mensajes 1 a 1 con contactos/usuarios: Se puede enviar un mensaje a cualquier usuario, proveyendo su nombre de usuario.
- Grupos: Se pueden crear grupos públicos, unirse a grupos, invitar a usuarios a un grupo, hablar en un grupo y aceptar automáticamente invitaciones a grupos.
- Status propio: Es posible modificar la disponibilidad, el modo y el mensaje de status propio del usuario.
- Notificaciones: Se reciben notificaciones de cambio de actividad, mensajes, solicitudes de amistad e invitaciones a grupos.
- Archivos: Se pueden enviar archivos de cualquier tipo a un usuario específico según su nombre de usuario. También se pueden recibir archivos y almacenarlos para que el usuario los pueda ver.

Dificultades

Tuve múltiples dificultades a lo largo del desarrollo del proyecto, las cuales se sobrepasé poco a poco y representaron retos interesantes.

Librería renovada

La librería que usé, Smack, en java, renovó diferentes funcionalidades hace poco. Lo cual me causó dificultades, porque al buscar en las fuentes de internet ejemplos y soluciones a problemas, solo hablaban de las versiones pasadas. Esto no me servía por la nueva versión de la librería. Además, la documentación no era muy buena. De hecho, la página donde se suponía que estaba la información no existe. Por lo que me tocó apoyarme de herramientas de IA y de buscar bastante en Stack Overflow y otros sitios.

Elección de lenguaje

La elección del lenguaje que iba a utilizar para desarrollar el proyecto fue un poco difícil. Primero dije que iba a usar GO para probar un nuevo lenguaje, pero no existen buenas librerías y era bastante complicado enviar las stanzas hechas a mano. Luego pensé en python pero no me decidí por la curva de aprendizaje de programación concurrente. Finalmente decidí usar Java, pero me topé con que tenía que crear un proyecto y usar maven para instalar la librería. Usar maven fue un reto porque nunca lo había usado. Usar maven en VSCode fue difícil, entonces me cambié a IntelliJ finalmente.

Tiempo

En ocasiones me pasa que trabajo por lotes, es decir, hago una buena cantidad de funcionalidades un día y dejo pasar tiempo hasta que vuelvo a hacerlo de nuevo. Pero en esta ocasión, como era necesario hacer uso adecuado del control de versiones, en ocasiones se me dificultó encontrar el tiempo adecuado para hacerlo. Además, ciertamente si tuve que trabajar por lotes en ocasiones.

Aprendizajes

Sobrepasar los retos me hizo tener distintos aprendizajes a lo largo del proyecto

Leer documentación

Leer la documentación de las librerías es algo un poco complicado, porque normalmente no es muy buena y es más fácil buscar ejemplos. Pero esta vez aprendí a buscar entender el uso correcto de la librería, por la falta de información, lo cual fue bastante beneficioso porque en el futuro intentaré dirigirme a la documentación antes que a ejemplos. Esto me permitirá usar mejor las herramientas.

Entender códigos ajenos

En ambientes controlados como la universidad, a veces es difícil exponerse a códigos ajenos. Pero en esta ocasión, como parte de mi investigación de la librería, tuve que ir a leer el código de esta en github. Esto es algo muy positivo, porque es muy útil para proyectos colaborativos en mi futuro laboral donde tendré que entender lo que otros programadores hacen. Además, también me gustaría participar en Open Source, por lo que esto me ayudará bastante.

Mejor organización de mi tiempo

Aprendí a distribuir mejor mis tareas de una forma más granular. En lugar de hacerlas todas de un solo, las dividí. Esto me ayudó a no saturarme tan rápidamente y a sentirme más productivo durante el desarrollo del proyecto.

Arquitectura de código

Aunque no lo mencioné como problema, sentí que la clase principal que maneja la conexión con el servidor se volvió demasiado grande. En ocasiones me perdía y confundía. Esto hará que en el futuro tome mejores decisiones de la arquitectura de mis proyectos y de la organización de mi código.