



C.P.R. Liceo “La Paz”

Proyecto Fin de Ciclo

Desarrollo de Aplicaciones Multiplataforma QuickChat

Autor: David García Soto

Tutor: Jesús Ángel Pérez-Roca Fernández

Resumen

El programa QuickChat es una aplicación web que está diseñada para proporcionar una plataforma de mensajería instantánea que sea rápida.

Su propósito es permitir a los usuarios crear grupos de comunicación de una o mas personas de manera fácil y eficiente.

Abstract

The QuickChat program is a web application designed to provide a fast instant messaging platform. Its purpose is to allow users to easily and efficiently create communication groups of one or more people.

Palabras Clave

- **JavaFX:** Es un conjunto de herramientas y bibliotecas de software desarrollado por Oracle para la creación de aplicaciones de escritorio y aplicaciones móviles modernas en Java.
- **MySQL:** Es un sistema de gestión de base de datos relacional ampliamente utilizado en el desarrollo de aplicaciones web y empresariales.
- **Hibernate:** Es un marco de trabajo de mapeo objeto-relacional para la plataforma Java.
- **TCP:** Es un protocolo de comunicación de red utilizado en Internet y en redes locales para proporcionar una comunicación fiable.
- **Threads:** Es una herramienta fundamental para el desarrollo de aplicaciones concurrentes y paralelas.
- **mensajería Instantánea:** Permite a los usuarios enviar mensajes de texto de forma rápida y segura a través de Internet.
- **Grupos:** Los usuarios pueden crear grupos de chat para comunicarse con múltiples personas al mismo tiempo.

Agradecimientos a todos los profesores que he tenido durante estos cursos.

Sumario

Resumen	3
Abstract.....	4
Palabras Clave	5
Introducción/motivación.	11
Objetivos.....	12
Estado del arte.....	13
Caso de estudio.....	14
Diagramas.	15
Desarrollo del proyecto	16
Manual Administrador	17
Manual Usuario.....	18
Viabilidad tecno-económica.	19
Trabajo futuro.....	20
Conclusiones.	21
Biblioteca de recursos web y referencias.....	22
Anexos.....	23

Introducción/motivación.

Las razones por lo que quise hacer este proyecto y los objetivos que tenía para lograr son estos(unifico la introducción/motivación con los objetivos):

- Porque cuando abarcamos el tema de los chats donde juntando el tema de los hilos más el tema de las conexiones TCP se podía hacer un chat donde se podían unir uno o más usuarios me parecía algún muy divertido de intentar con una interfaz gráfica y me parecía un muy buen reto para intentar hacer para el proyecto.
- Unos de objetivos que quería conseguir con esta aplicación es hacerla mas intuitiva para el usuario para que sepa en todo momento en donde se encuentra sin tener que pensar apenas.
- Otro objetivo era conseguir maximizar los recursos y conseguir la máxima eficiencia al mandar los mensajes.
- El ultimo objetivo era poder crear dos o mas grupos a la vez y que el cliente se puede pasar de uno a otro sin tener ningún tipo de problema.

Objetivos.

Estado del arte.

IMessage:

- Un punto fuerte de esta aplicación es su función de 'Apple pay' donde permite a los usuarios enviar y recibir dinero fácilmente.
- Un punto débil de esta aplicación es su ecosistema cerrado donde falta la compatibilidad con otros sistemas operativos.

PS App:

- Un punto fuerte de esta aplicación es su función de crear comunidades donde puede haber hasta mas de 20000
- Un punto débil de esta aplicación es el rendimiento de esta, ya que muchas veces los mensajes tardan en llegar o no se llegan a enviar.

Las tecnologías mas usadas en mi aplicación son Java que es un lenguaje de programación orientada a objetos, JavaFX que es un conjunto de herramientas y bibliotecas de software desarrollado por Oracle para la creación de aplicaciones y MySQL Es un sistema de gestión de base de datos relacional.

Otras alternativas que pude utilizar por ejemplo serian MongoDB en la parte de sistema de gestión de base de datos y kotlin que es un lenguaje de programación mas moderno que Java.

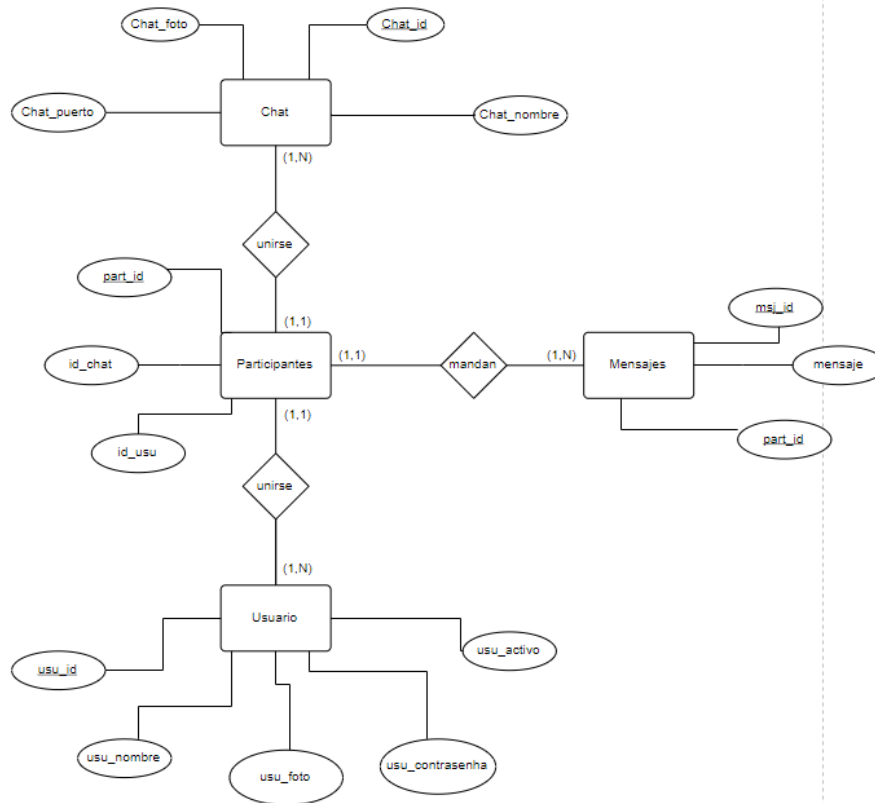
Caso de estudio.

En la aplicación QuickChat se han desarrollado varios puntos donde se soluciona problemáticas dentro de la aplicación de mensajería instantánea.

- **Menú sencillo:**
 - En QuickChat hemos querido reducir el numero de botones que habitualmente tiene los menus de mensajería instantánea ayudando al usuario a ver claramente los diferentes apartados dentro de la aplicación.
- **Rapidez en la mensajería:**
 - En QuickChat no hemos centrado en querer reducir al mínimo los recursos utilizados y mejorar la rapidez de los mensajes.

Diagramas.

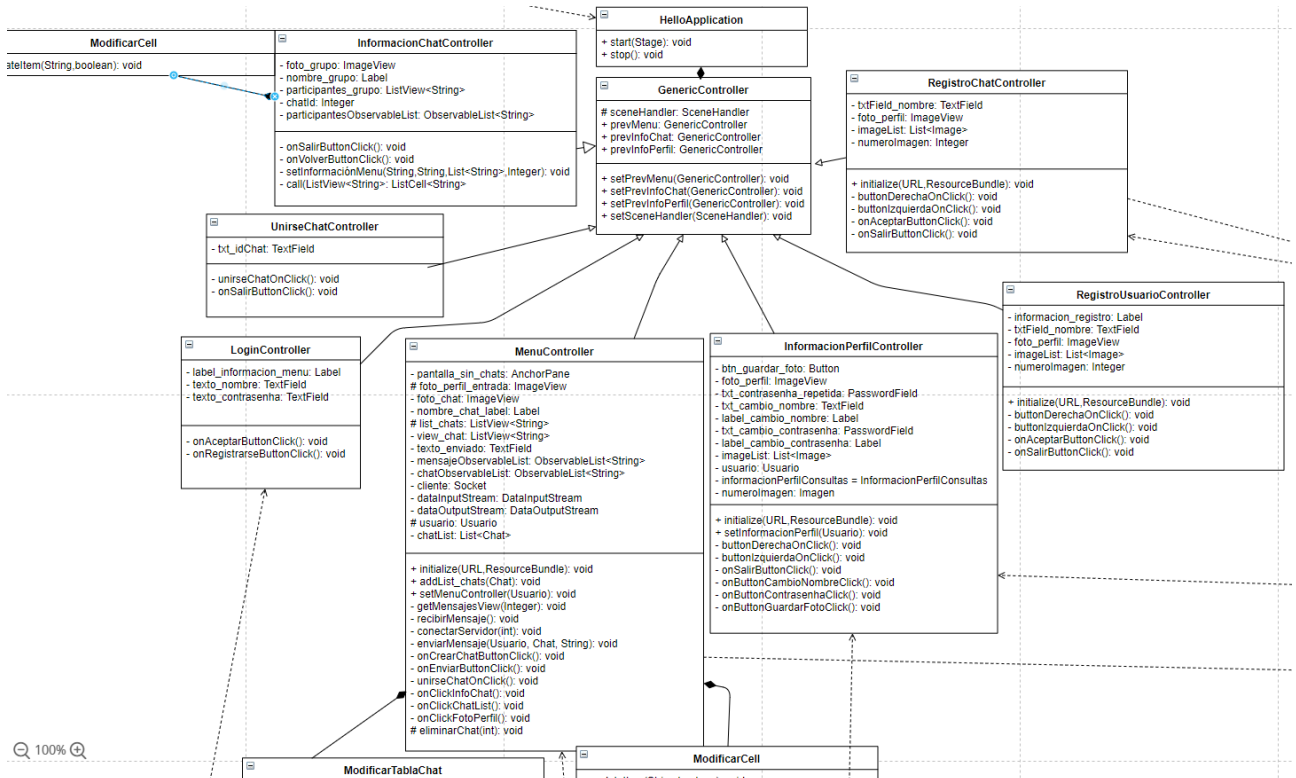
Explicación diagrama E/R (Entidad Relación):



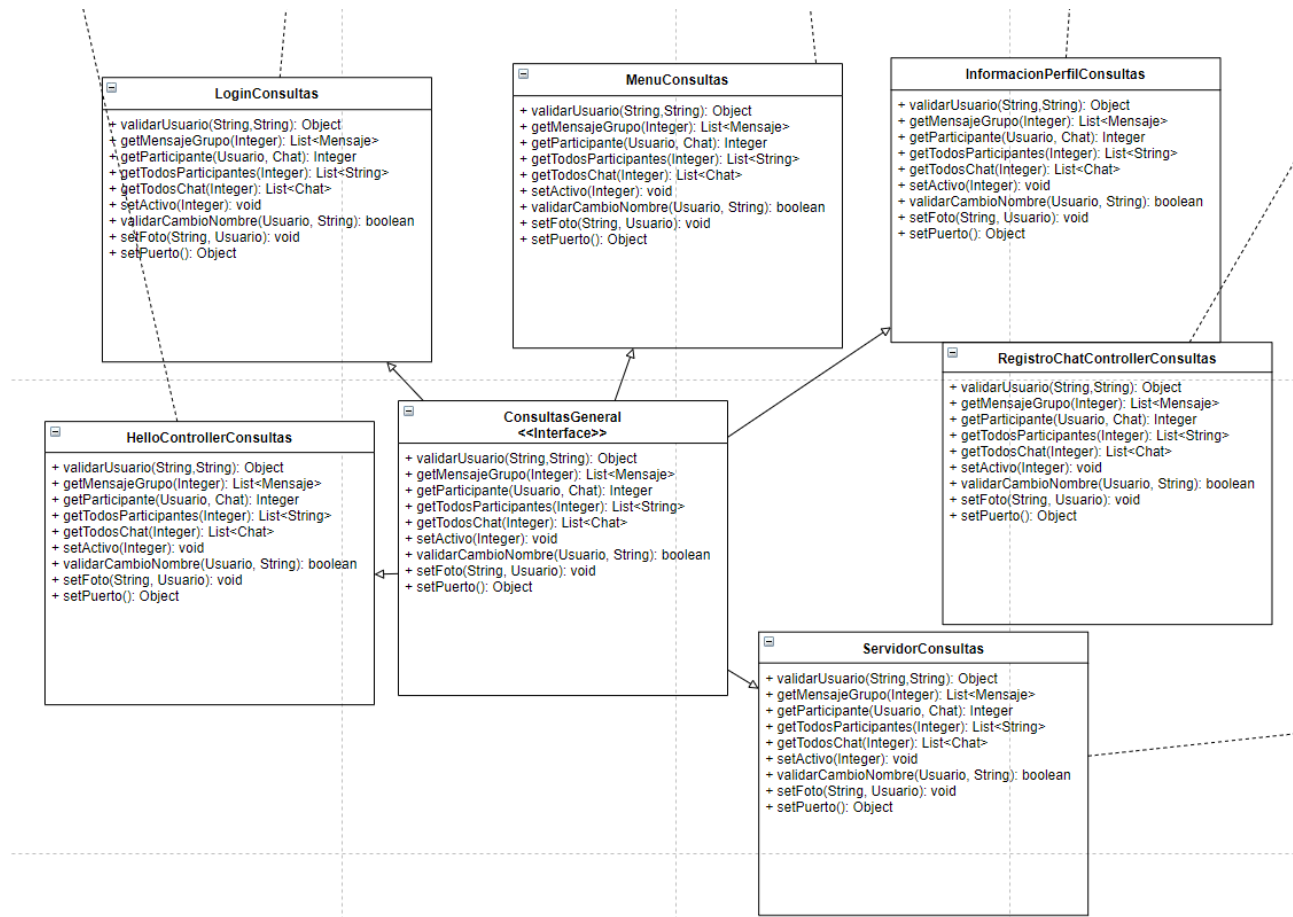
- **Chat:** Representa un chat, que puede tener múltiples participantes. Cada chat tiene atributos como foto, puerto, nombre e identificador único.
- **Participantes:** Actúa como una entidad intermedia que conecta los chats con los usuarios. Cada participante tiene su propio identificador y referencias a un chat y un usuario.
- **Usuario:** Representa un usuario del sistema de chat, con atributos como nombre, foto, contraseña y estado activo.
- **Mensajes:** Representa los mensajes enviados dentro de los chats. Cada mensaje tiene un identificador único, contenido del mensaje y una referencia al participante que lo envió.

Explicación diagrama de clases:

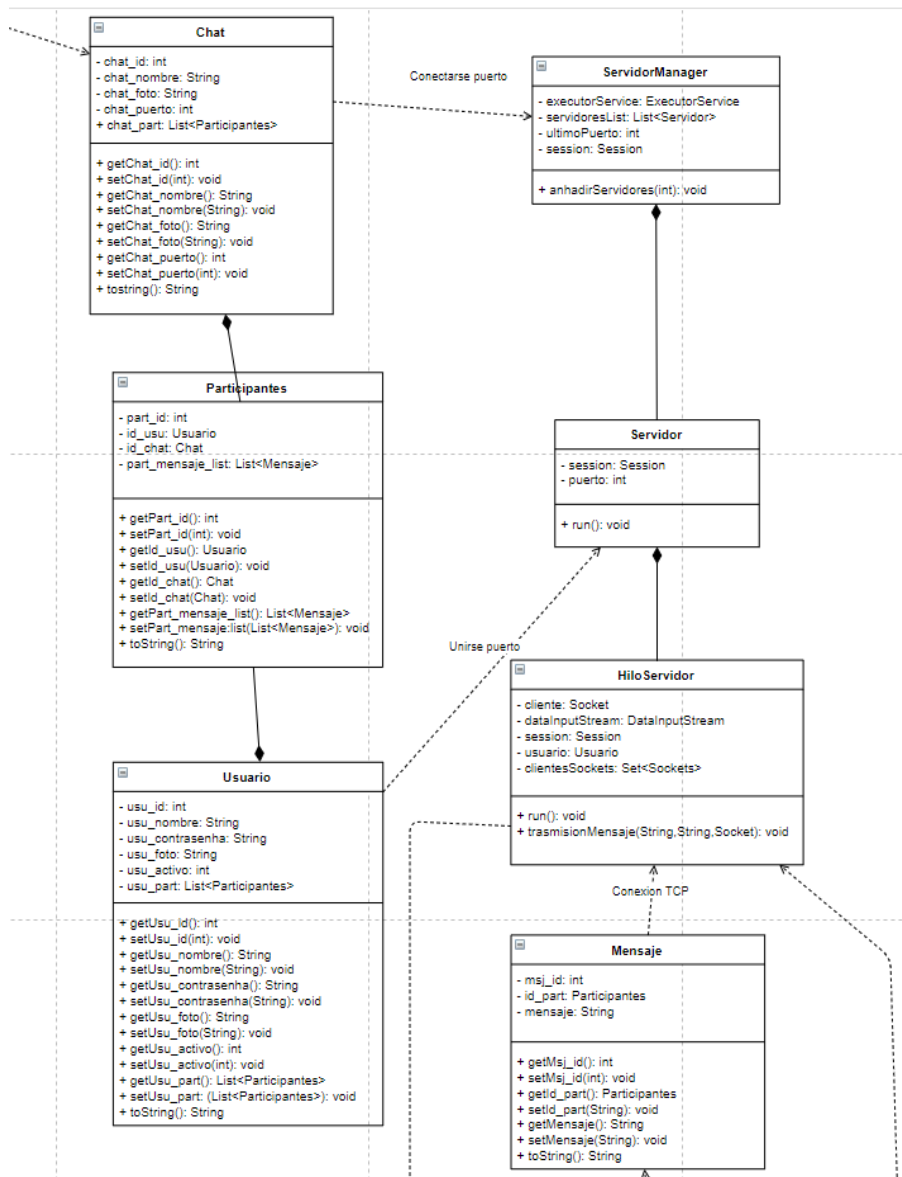
Se ha tenido que dividir en cuatro fotos ya que no entraba todo en una foto.



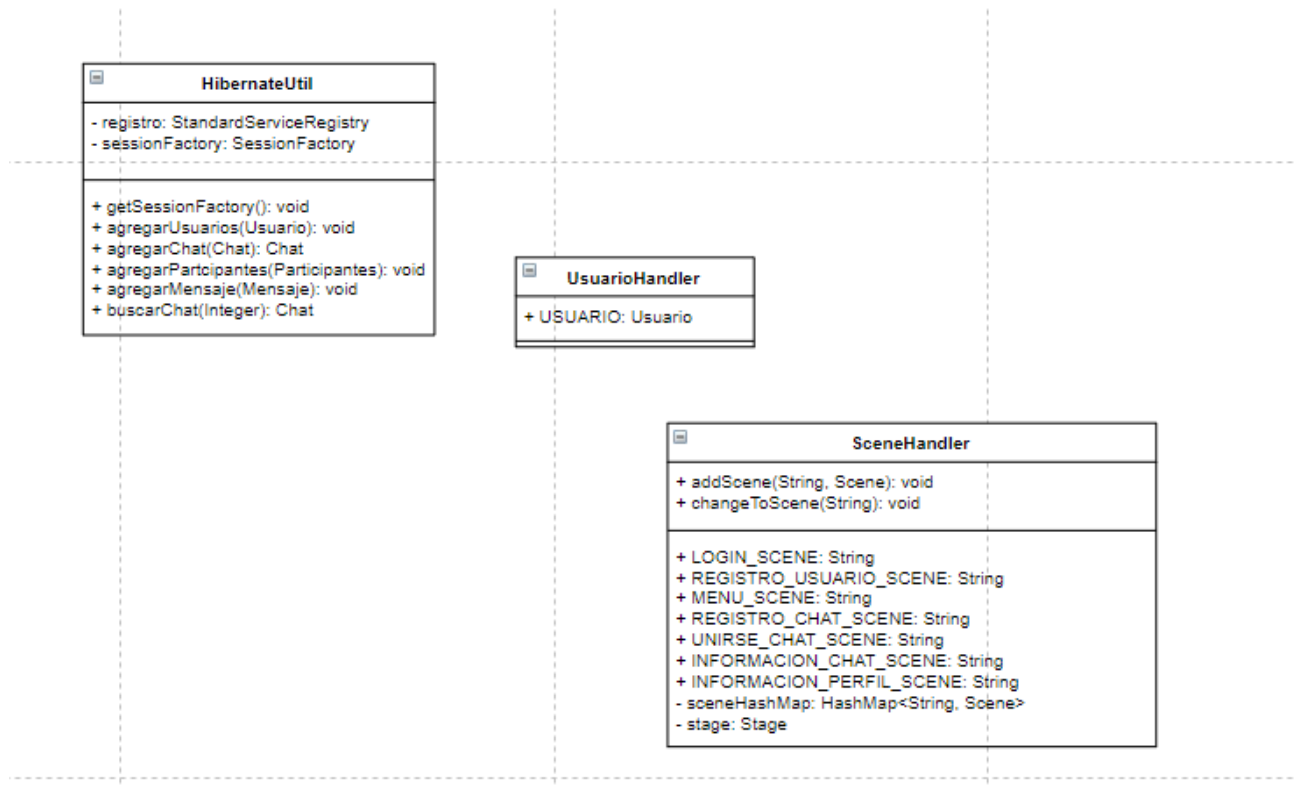
En esta primera foto podemos apreciar todos los controllers de la aplicación que tanto MenuController e InformacionChatController tienen clases de composición. Todas estas clases extienden de GenericController y esta tiene una relación de composición con HelloApplication.



En esta segunda foto se puede apreciar todas las clases donde se ejecuta consultas del programa y que cada una esta relacionada con los controller de la anterior foto. Todas extienden de la interface ConsultasGeneral.

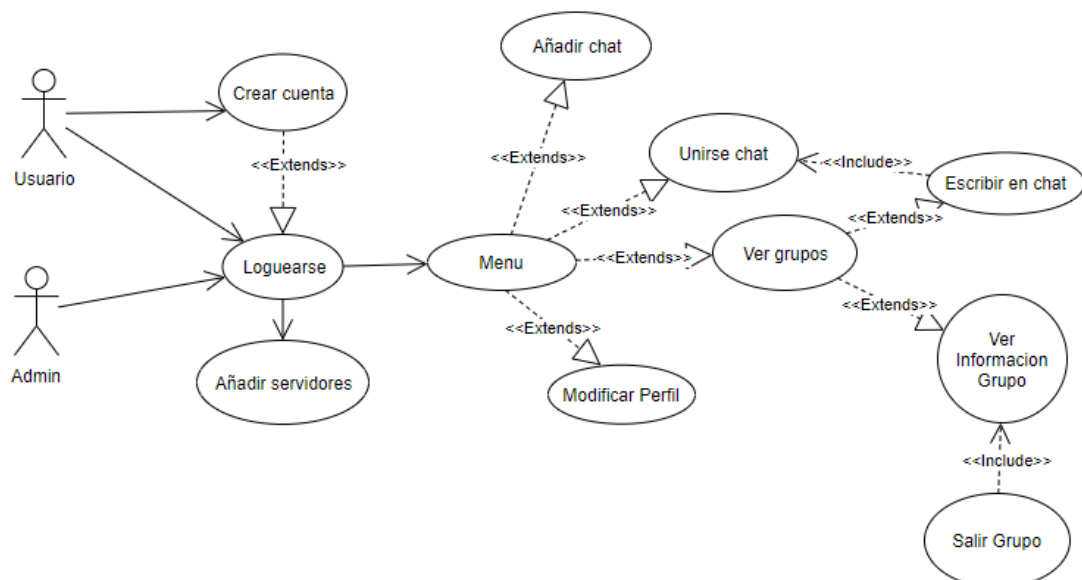


En esta tercera foto se puede apreciar el sistema de mensajería instantánea y las entidades del programa donde cada una se relaciona dependiendo de las conexiones TCP que hay entre ellas o si se tienen que conectar a un puerto o a otro.



En esta ultima foto se puede apreciar las clases restantes que no tiene relación con ninguna de las otras.

Explicación del diagrama de casos de uso:



Se puede observar que hay dos actores, el usuario y el admin. El admin primero necesita logearse para después poder añadir servidores. El cliente puede o crearse una cuenta o logearse directamente para pasar al menú donde hay puede escoger este o añadir un chat, unirse a un chat, ver los grupos en los que esta o modificar su perfil.

El usuario si quiere escribir un mensaje va a necesitar primero unirse a uno si es que no esta en ninguno o meterse en el grupo directamente y si se quiere salir del grupo necesita entrar en los detalles del grupo.

Desarrollo del proyecto

Primero antes de empezar el desarrollo de la aplicación definí los requisitos que creía mas importantes para la aplicación e identifiqué las tecnologías que iba a utilizar.

Después de tener claro los requisitos más importantes e identificar las tecnologías, pase al diseño de la base de datos para dejar planificada la estructura principal de la aplicación para cuando se empiece el desarrollo ya tener una estructura fija.

Lo siguiente es empezar a desarrollar la aplicación, esta dentro del menú tiene varias opciones:



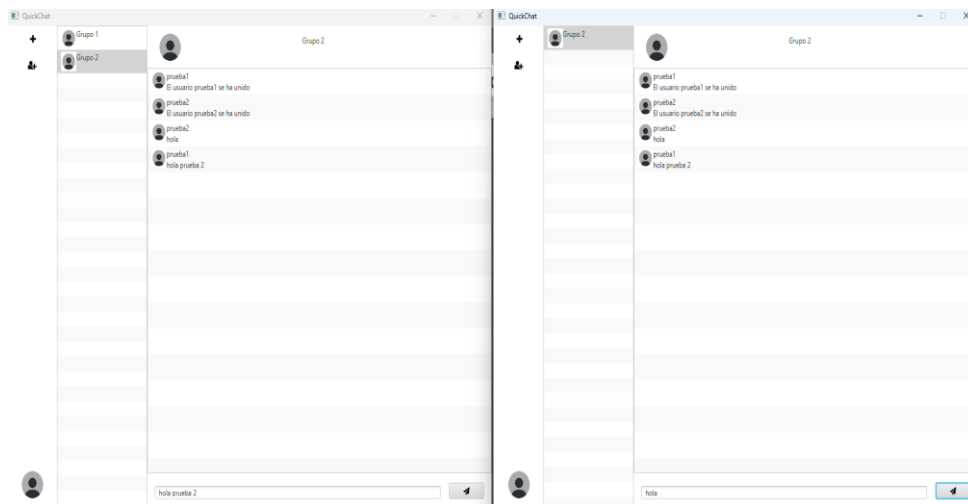
La primera opción es poder crear un chat nuevo. Dentro de esta opción la aplicación te dejara escoger la foto para el grupo dentro de unas opciones limitadas y el nombre del grupo.

La segunda opción es poder unirse a un grupo. Dentro de esta opción el cliente podrá poner el id del grupo para poder unirse al grupo seleccionado.

La última opción es poder modificar tu nombre, la foto y la contraseña.



En esta captura se ve que se puede tener 1 o mas grupos a la vez y los clientes pueden ir cambiando de grupo cuando quieran. Esto se consiguió gracias a el servidor mánager que crea servidores para que se puedan conectar los grupos que sean creados por los clientes.



En esta captura se puede ver que los mensajes llegan a los clientes, esto ocurre ya que hay un servidor donde se conectan estos clientes, los clientes mandan un mensaje que se envía a través del protocolo TCP donde a parte se manda como información para el servidor quien envió el mensaje y el servidor se encarga de recoger esos datos y enviar el mensaje a todos los clientes conectados a ese grupo.

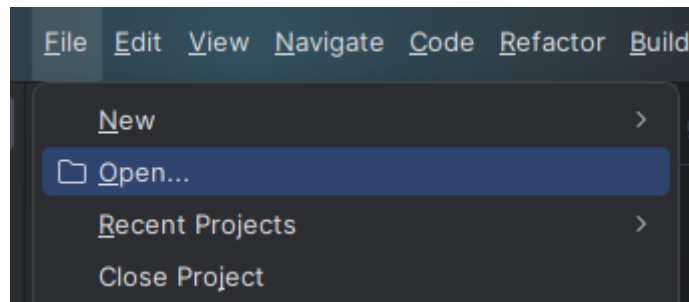
Por ultimo fue hacer el diseño de la aplicación y arreglar los últimos bugs para dejarla lista para entregar.

Manual Administrador

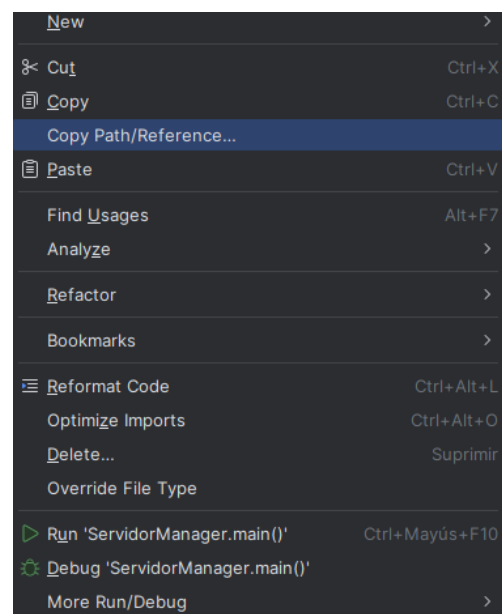
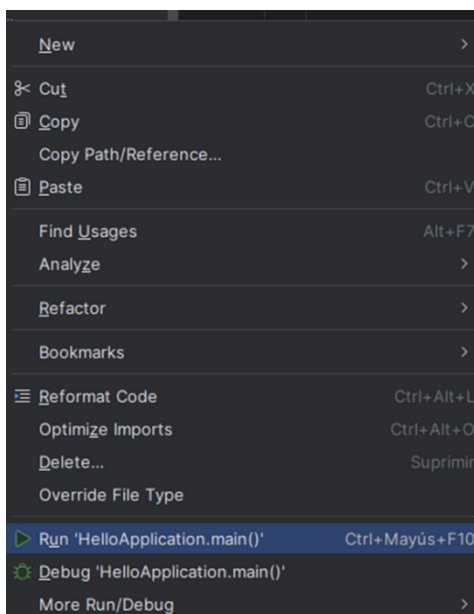
Requisitos del sistema:

- Un entorno de desarrollo integrado (IntelliJ IDEA)
- JDK 18
- Ver 8.0.27 de MySQL

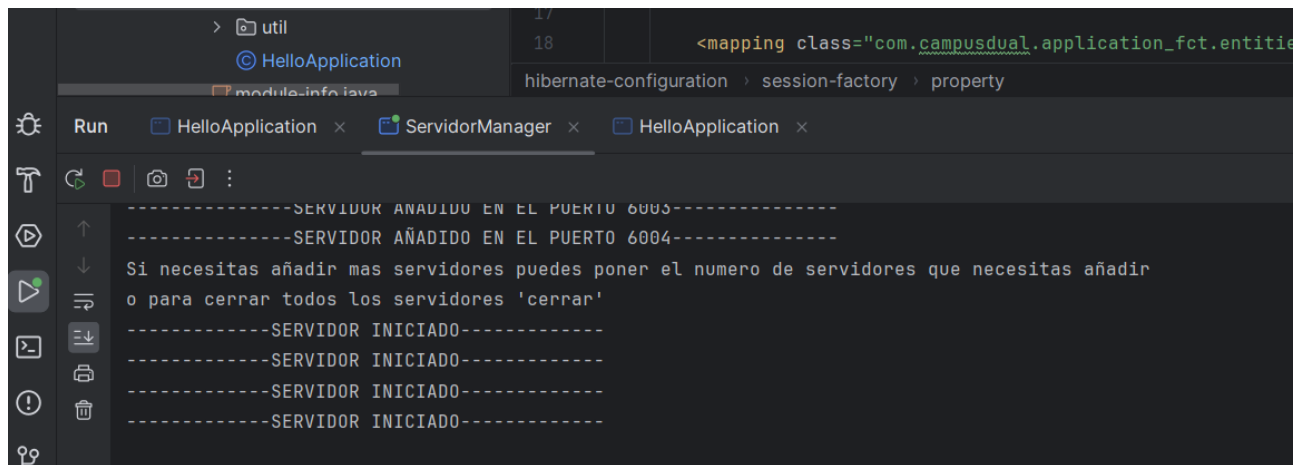
Proceso de instalación:



- En primer lugar entras en la aplicación de IntelliJ IDEA y vas a la pestaña File y le das a Open y buscas el proyecto dentro de tus archivos.



- En segundo lugar cuando el programa cargue del todo te tendrás que mover por la carpeta src/main/java/com.campusdual.application_fct/servicio y le dará click derecho a la clase ServidorManager. Si quiere el administrador utilizar el programa hará lo mismo pero con la clase HelloApplication



```
util
  HelloApplication
    module-info.java
      hibernate-configuration > session-factory > property
17
18
    <mapping class="com.campusdual.application_fct.entitie
hibernate-configuration > session-factory > property

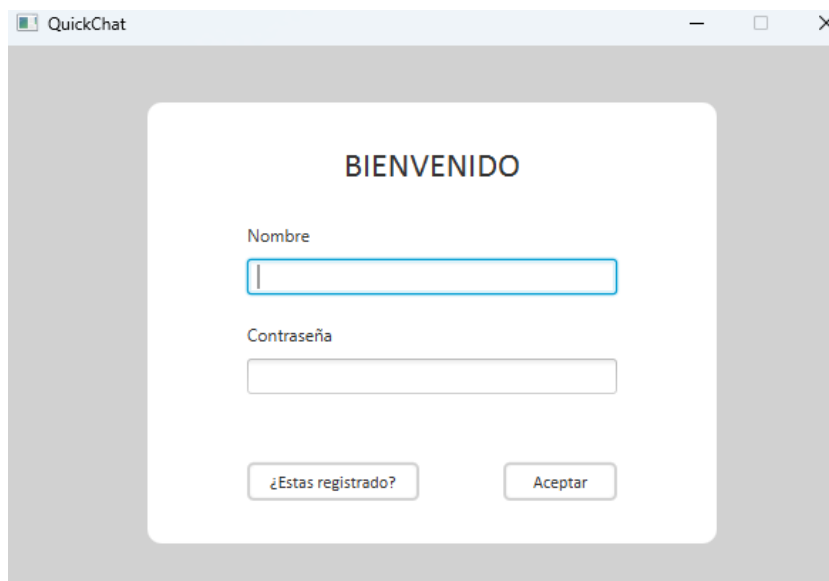
Run HelloApplication x ServidorManager x HelloApplication x

-----SERVIDOR AÑADIDO EN EL PUERTO 6003-----
-----SERVIDOR AÑADIDO EN EL PUERTO 6004-----
Si necesitas añadir mas servidores puedes poner el numero de servidores que necesitas añadir
o para cerrar todos los servidores 'cerrar'
-----SERVIDOR INICIADO-----
-----SERVIDOR INICIADO-----
-----SERVIDOR INICIADO-----
-----SERVIDOR INICIADO-----
```

- Si el administrador necesita iniciar mas servidores lo único que tendrá que hacer será poner el numero de servidores que necesita. Por ejemplo y pone 2 se iniciaran 2 servidores.

Manual Usuario

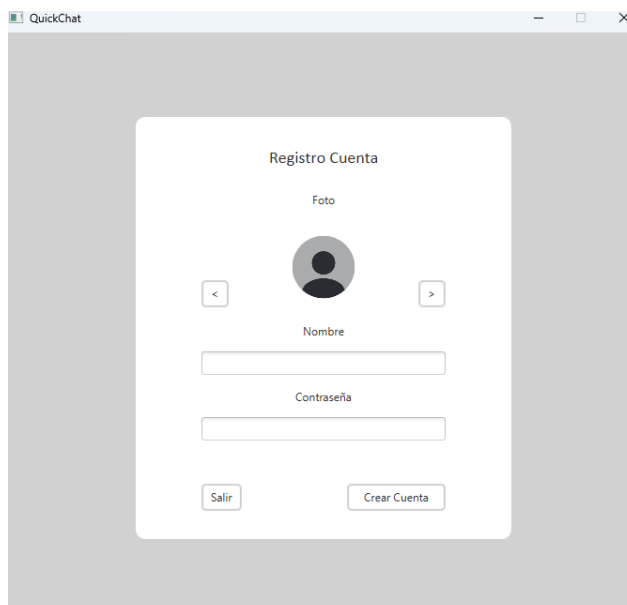
Inicio de la aplicación:



The screenshot shows a window titled 'QuickChat' with a login form. The form has a title 'BIENVENIDO'. Below the title are two input fields: 'Nombre' (Name) and 'Contraseña' (Password). At the bottom of the form are two buttons: '¿Estas registrado?' (Are you registered?) and 'Aceptar' (Accept).

- En esta vista podemos tendremos que poner el nombre y la contraseña que se tenga registrada en la aplicación. En el caso de que se ponga una algún campo mal y se le de al botón 'aceptar' saldrá un error dependiendo de el fallo, sino se entrara dentro del programa, en el caso de no tener cuenta habrá que darle al botón de '¿Estas registrado?'

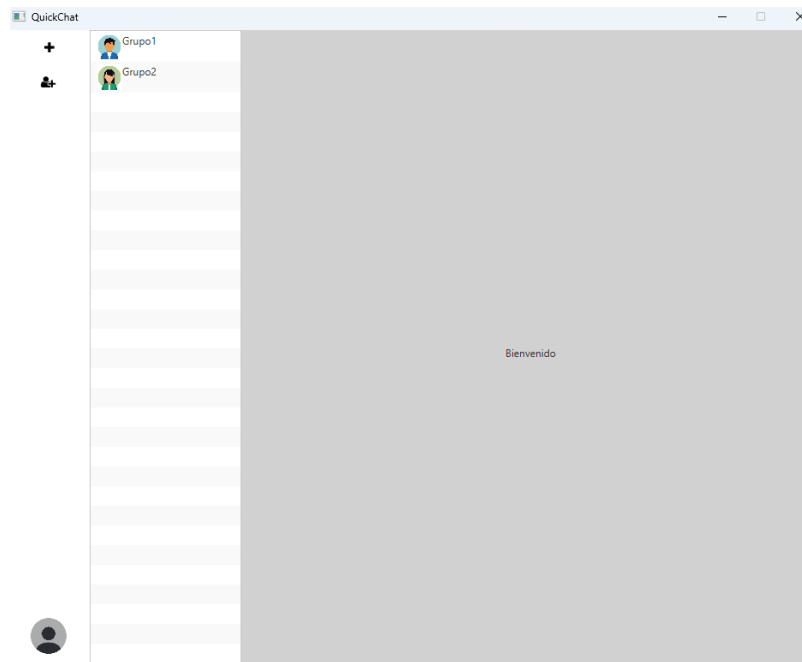
Vista de registro de usuario:



The screenshot shows a window titled 'QuickChat' with a user registration form. The form has a title 'Registro Cuenta'. Below the title is a 'Foto' (Photo) section with a circular placeholder and navigation arrows. Below the photo are two input fields: 'Nombre' (Name) and 'Contraseña' (Password). At the bottom of the form are two buttons: 'Salir' (Exit) and 'Crear Cuenta' (Create Account).

- En esta vista podremos registrarnos, podrás escoger una foto dentro de las fotos que te da a escoger el programa. Si intentas poner un nombre que ya está registrado no te dejara registrarse, sino se creara la cuenta al darle al botón de crear cuenta, si quieres salir de la vista tendrás que darle al botón de salir.

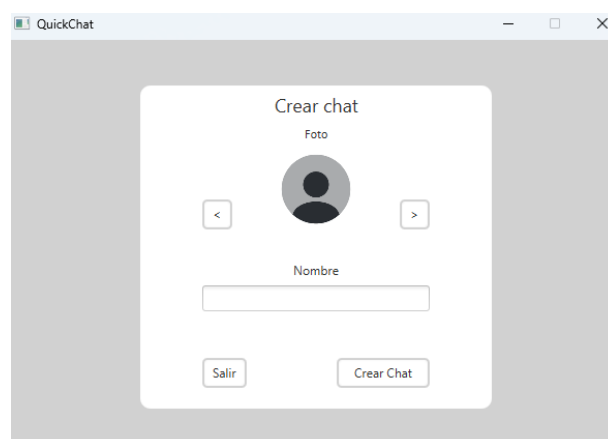
Vista menú principal:



- En esta vista podremos ver varias cosas, entre ellas un menú lateral donde hay 3 botones:
 - El icono del mas que te lleva a una vista para crear un grupo.
 - El icono de una persona y un mas que te lleva a una vista para unirse a un grupo.
 - Tu foto de perfil que te lleva a una vista donde podrás modificar tu perfil.

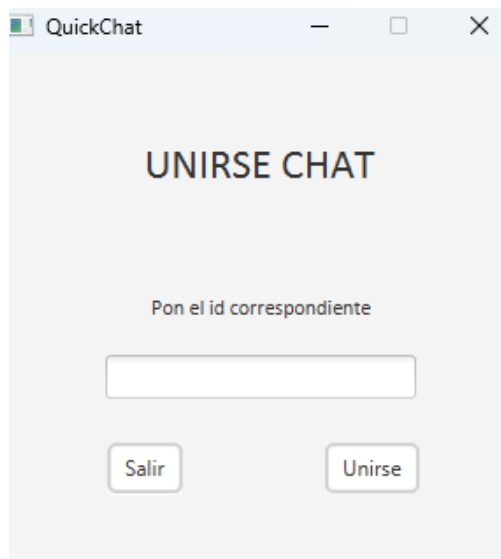
También se puede ver si estas en algún grupo o no.

Vista para crear un chat:



- En esta vista podremos ver la opción de crear un grupo donde te da la opción para poner una foto dentro de las opciones de la aplicación y el nombre del grupo.

Vista para unirte a un chat:



QuickChat

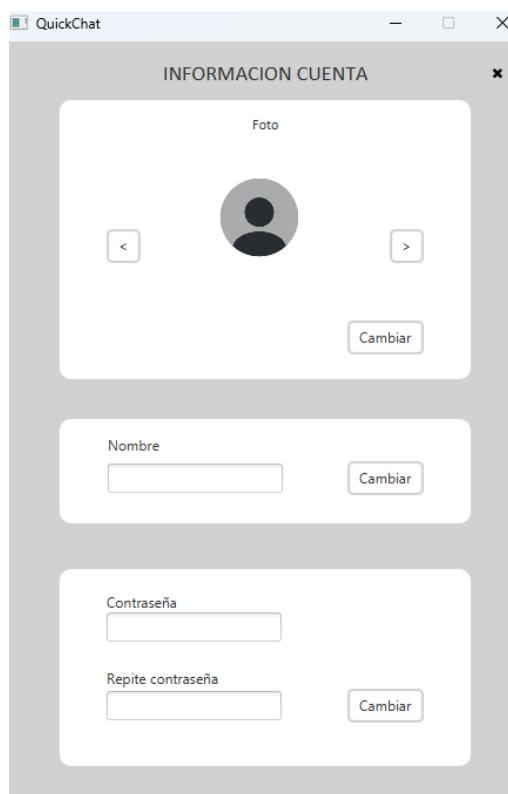
UNIRSE CHAT

Pon el id correspondiente

Salir Unirse

- En esta vista podremos ver la opción de unirse a un chat donde tendrás que poner el id del grupo para poder unirte al grupo, si pones un id que no existe te salta un aviso donde pone que no existe ese grupo, si todo fue bien al darle al botón 'unirse' se unirá al grupo seleccionado.

Vista de modificación de perfil:



QuickChat

INFORMACION CUENTA

Foto

< >

Cambiar

Nombre

Cambiar

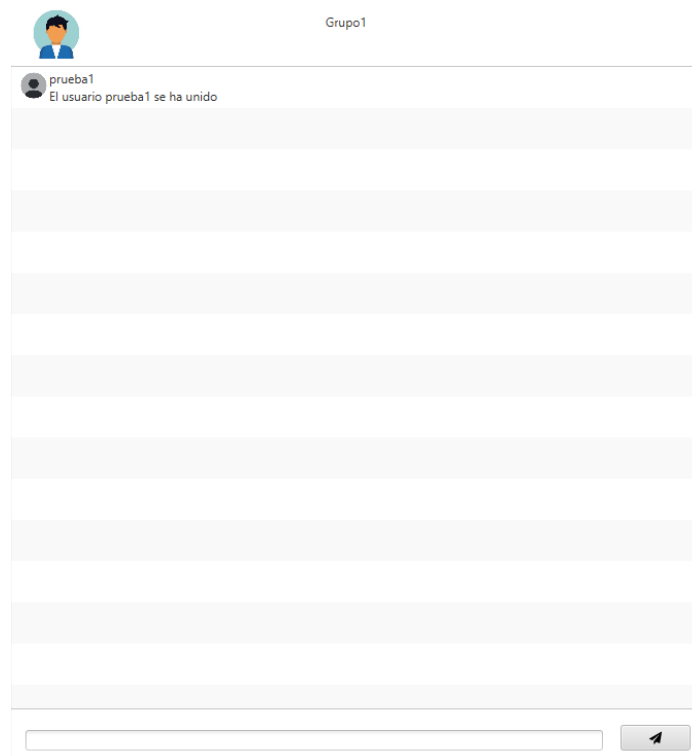
Contraseña

Repite contraseña

Cambiar

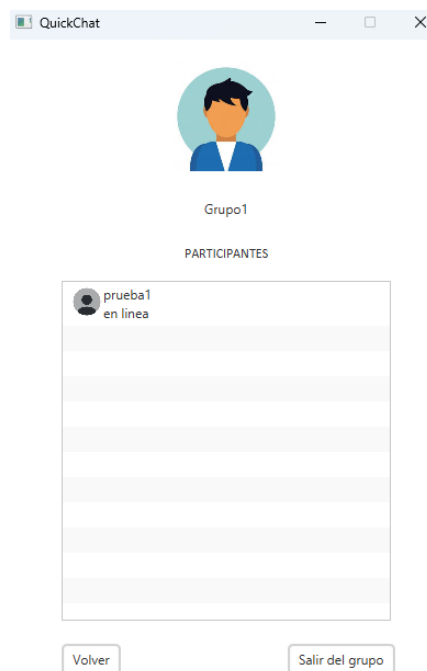
- En esta vista podremos ver las diferentes opciones para poder modificar tu perfil, introduces al campo mal te saldrá el error dependiendo de lo que has hecho mal.

Vista dentro de un grupo:



- En esta vista podremos ver los mensajes que se han escrito en el grupo, para escribirlos tu que para mandarlos tendréis que darle al botón de enviar. Si se quiere ver los detalles del grupo tendréis que darle donde esta la foto de grupo o a el nombre.

Vista de los detalles del grupo:



Proyecto Desarrollo de Aplicaciones Multiplataforma

- En esta vista se podrá ver los detalles del grupo como su foto, su nombre y los participantes de este grupo, también se podrá ver si estos están en línea o no. Si quieres volver al menú será en el botón de 'volver' y si quieres salir del grupo será en el botón de 'salir del grupo'.

Viabilidad tecno-económica.

Costes del Hardware:

- Portátil:
 - Teniendo en cuenta las especificaciones medias para poder programar sin ningún tipo de problema tendremos en cuenta que el portátil tendrá:
 - Procesador (CPU): Intel Core i5.
 - Memoria RAM: 16 GB.
 - Almacenamiento 512 GB SSD.
 - Tarjeta Gráfica (GPU): Integrada (Intel Iris Xe).
 - Sistema Operativo: Windows 10/11.

Esto hace que el portátil cueste sobre 1000 - 1500 euros.

- Monitor:
 - Teniendo en cuenta las especificaciones medias para un buen monitor tendrá:
 - Tamaño de pantalla: 24 a 27 pulgadas.
 - Resolución: Full HD (1920X1080).

Esto hace que el monitor cueste sobre las 200 – 300 euros.

- Periféricos:
 - Teniendo en cuenta periféricos medios para que la experiencia sea mejor el precio será sobre unos 50 euros.

Costes del Software:

- IntelliJ IDEA (versión gratuita):
 - El uso de la versión gratuita de IntelliJ IDEA elimina cualquier costo asociado con el IDE. Esta versión proporciona todas las herramientas esenciales para el desarrollo de aplicaciones Java, incluyendo soporte para JavaFX y bases de datos como MySQL.
- JavaFX:
 - JavaFX es una biblioteca de gráficos y medios de código abierto que se puede utilizar sin costo adicional. Es una opción viable para desarrollar interfaces de usuario ricas y modernas en aplicaciones Java.

- MySQL (versión gratuita):
 - MySQL es un sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado y disponible sin costo adicional. Proporciona una solución robusta y escalable para las necesidades de bases de datos del proyecto.
- JDK (Java Development Kit):
 - El JDK necesario para compilar y ejecutar aplicaciones Java está disponible gratuitamente a través de Oracle OpenJDK.
- Sistema Operativo:
 - Asumiendo que el portátil viene con un sistema operativo preinstalado (Windows, macOS, o una distribución gratuita de Linux), no se incurren costos adicionales en este aspecto.

Otros Costes:

- Gastos de mantenimiento y soporte: Aunque el hardware y software utilizados no requieren costos iniciales elevados, es prudente considerar los gastos de mantenimiento y soporte técnico que pueden surgir a lo largo del tiempo. Esto puede incluir actualizaciones de software, reparaciones de hardware y otros imprevistos.
- Formación y capacitación: Si el equipo de desarrollo no está familiarizado con IntelliJ IDEA, JavaFX o MySQL, podría ser necesario invertir en formación. Sin embargo, existen numerosos recursos gratuitos en línea que pueden minimizar estos costos.

Rentabilidad del Proyecto:

- La rentabilidad del proyecto se puede analizar desde dos perspectivas: costos iniciales y beneficios a largo plazo.
 - Costos iniciales:
 - Portátil: 1000 - 1500 euros
 - Monitor: 200 - 300 euros
 - Periféricos: sobre unos 50 euros
 - Software: 0 euros (versión gratuita de IntelliJ IDEA, JavaFX, MySQL y JDK)

Proyecto Desarrollo de Aplicaciones Multiplataforma

- Utilizando los precios mas caros el total seria de 1850 euros
- Beneficios a largo plazo:
 - Al utilizar herramientas y recursos gratuitos, se minimizan los costos operativos.
 - El uso de un portátil acompañado de un monitor y periféricos, garantiza un entorno de desarrollo cómodo y eficiente sin necesidad de inversiones adicionales a corto plazo.
 - La capacidad de desarrollar aplicaciones ricas en interfaz de usuario con JavaFX y manejar bases de datos con MySQL puede atraer a una mayor base de usuarios y potencialmente aumentar los ingresos del proyecto.

Conclusión:

La implementación del proyecto utilizando IntelliJ IDEA (versión gratuita), JavaFX, MySQL y el hardware descrito es tecnológicamente viable y económicamente rentable. Los costos iniciales son relativamente bajos y, gracias a las herramientas gratuitas y los recursos disponibles, el proyecto puede desarrollarse y mantenerse con un presupuesto ajustado. La rentabilidad del proyecto dependerá de la calidad del software desarrollado y de su aceptación en el mercado, pero desde un punto de vista tecno-económico, se trata de una inversión prudente y prometedora.

Trabajo futuro.

Esta aplicación se le podrían implementar muchas funcionalidades, pero las más importantes serían:

- **Añadir función contactos:** La aplicación tendría una pestaña donde se guardarían sus contactos y el cliente podría acceder a ellos más fácilmente a parte de poder añadir nuevos o borrarlos.
- **Mandar archivos en los grupos:** La aplicación tendría un botón al lado de el botón de enviar mensaje donde podría adjuntar un archivo y poder enviarlo por el grupo.
- **Poder llamar al grupo o a tus contactos:** La aplicación tendría un botón en la información del grupo donde podrías llamar a todo el grupo o poder llamar individualmente a los contactos que tengas agregados.
- **Mejoras de personalización:** En la aplicación podrías escoger la foto de perfil de tu galería y hacer lo mismo con las fotos de los grupos.
- **Mejoras de seguridad:** La aplicación la única medida de seguridad que tiene es la encriptación de contraseña, pero medidas de seguridad para otro tipo de métodos de hackeo.
- **Poder enviar audios por los grupos:** En la aplicación tendrías la opción de poder mandar un audio a los grupos en vez de tener solo la opción de escribir.

Conclusiones.

Explicar si se han obtenido los objetivos propuestos al principio del proyecto y si no se han conseguido algunos explicar las razones por las que no se pudieron cumplir, explicar posibles líneas de investigación futura para añadir funcionalidades, mejorar el aspecto gráfico, el rendimiento, etc.

Como conclusión me ha parecido un proyecto bastante divertido pero tuve muchas complicaciones con el tema de los servidores ya que no conseguía tener mas de un servidor al mismo tiempo y que el cliente se uniera a cualquiera de estos sin ningún tipo de problema, pero al final después de mucho tiempo y esfuerzo conseguí hacerlo.

Como posibles líneas de investigación para futuras funcionalidades está la de unirse a un chat sin tener que pasarle el id, sino invitando al usuario que quieras el otro usuario, otra funcionalidad seria la de poder mandar fotos y videos.

Biblioteca de recursos web y referencias.

JavaFX, I. (2014): JavaFX API, Oracle, disponible en:

<https://docs.oracle.com/javase/8/javafx/api/overview-summary.html>

JavaFX, I. (2023): ListView JavaFX 20, Oracle, disponible en:

<https://openjfx.io/javadoc/20/javafx.controls/javafx/scene/control/ListView.html>

GutHub, I. (2017): fontawesomefx-glyphsbrowser, disponible en:

<https://github.com/Jerady/fontawesomefx-glyphsbrowser>

Libro, I. (2022): programación de servicios y procesos, Paraninfo

MvnRepository, I.(2024): fontAwesomeFx, MVNRepository, disponible en:

<https://mvnrepository.com/artifact/de.jensd/fontawesomefx-fontawesome>

Anexos.