

TECNOLOGÍA DEL DESARROLLO SOFTWARE

PROYECTO “AppChat”



Alumnos, DNI, Correo:

Mohammed Amrou Labied Nasser 49857930W

ma.labiednasser@um.es

Sergio García García 49308323A

sergio.garciag1@um.es

Grupo y Curso: Tercero, 1.1

Convocatoria: 2024-2025 enero

Tutor: José Ramón Hoyos Barceló

INDICE

| | |
|---|----|
| Historia de Usuario: | 3 |
| Actividad 1: Login y Registro de usuarios..... | 3 |
| Actividad 2: Gestión de contactos | 3 |
| Actividad 4: Gestión de Mensajes..... | 4 |
| Actividad 5: Búsqueda de mensajes | 5 |
| Actividad 6: Funcionalidad Premium | 5 |
| Diagrama de Clases de Dominio: | 6 |
| Diagrama de Operación de Añadir Contacto a Grupo: | 7 |
| Arquitectura de la App:..... | 8 |
| Decisiones de interés para comprensión: | 9 |
| Patrones de Diseño Usados: | 10 |
| Manual de Usuario: | 11 |
| Registrarse como Usuario: | 11 |
| Inicio de Sesión: | 12 |
| Ventana Principal de la Aplicación: | 12 |
| Agregar/Modificar Contacto:..... | 13 |
| Abrir Conversación: | 15 |
| Crear/Modificar Grupo: | 15 |
| Buscar Mensajes: | 17 |
| Funcionalidades Premium:..... | 20 |
| Recibir Mensajes: | 22 |
| Observación Final: | 22 |

Historia de Usuario:

Según las condiciones expresadas por el cliente, hemos recogido las siguientes historias de usuario que expresan el objetivo, algunos requisitos y los criterios de aceptación.

Actividad 1: Login y Registro de usuarios

Objetivo: Permitir a los usuarios acceder a la aplicación y sus funcionalidades después de registrarse y autenticarse.

Historias de Usuario:

Historia 1.1: Registro de Nuevo Usuario

-Como usuario, quiero registrarme proporcionando mi nombre completo, fecha de nacimiento, email, imagen de perfil, número de teléfono, contraseña y un mensaje de saludo opcional para acceder a la aplicación.

-Criterios de Aceptación: Todos los campos obligatorios deben completarse correctamente (menos la fecha de nacimiento y la imagen) y recibir un mensaje de confirmación tras el registro. Debe mostrarse un mensaje de error si algún campo es incorrecto o está incompleto. Además, la contraseña debe ser rellenada dos veces coincidiendo ambas y tener como mínimo 8 caracteres.

Historia 1.2: Inicio de sesión

-Como usuario, quiero iniciar sesión con mi número de teléfono y contraseña para acceder a mi cuenta.

-Criterios de Aceptación: Puedo iniciar sesión si las credenciales son válidas o recibir un mensaje de error si el usuario no está previamente registrado. Si las credenciales son correctas, debo ser redirigido a la pantalla principal de la aplicación.

Actividad 2: Gestión de contactos

Objetivo: Permitir al usuario poder registrar a nuevos contactos.

Historias de Usuario:

Historia 2.1: Registrar y eliminar contacto

-Como usuario, quiero registrar a un contacto proporcionando un nombre, un teléfono y una imagen para poder enviar o recibir mensajes. Además de ser capaz de eliminarlo.

-Criterios de Aceptación: Puedo registrar a un contacto si su número de teléfono es correcto o recibir un mensaje de error si el número de teléfono no existe. Si el número de teléfono es correcto podré enviar mensajes. Podré ser capaz de eliminar un contacto si es previamente un contacto mío.

Historia 2.2: Foto de perfil

-Como usuario, quiero añadir o cambiar mi foto de perfil en cualquier momento para mejorar la interfaz.

-Criterios de Aceptación: Puedo cambiar mi foto de perfil si está en el formato correcto, en caso de no estar se mostrará un mensaje de error y no se realizará ningún cambio.

Actividad 3: Gestión de Grupos

Objetivo: Permitir al usuario poder crear grupos de contactos para facilitar el envío de mensajes a varios contactos ya agregados de manera simultánea.

Historias de Usuario:

Historia 3.1: Crear y eliminar grupo

-Como usuario, quiero ser capaz de poder crear o eliminar un grupo que solo tenga un nombre y opcionalmente agregarle una foto para posteriormente agregarle contactos al grupo.

-Criterios de Aceptación: Puedo crear grupos siempre que contenga uno o más contactos y agregarle un nombre al grupo y opcionalmente una foto. Si no se le agrega un nombre, se mostrará un mensaje de error y no permitirá la creación de esta. Además, puedo borrar un grupo siempre que exista.

Historia 3.2: Añadir y eliminar contacto a grupo

-Como usuario, quiero ser capaz de poder agregar o eliminar un contacto que ya tengo guardado a un grupo en el que pertenezco.

-Criterios de Aceptación: Puedo poder agregar o eliminar un contacto de un grupo existente, dará un mensaje de error si se intenta agregar un grupo a otro grupo.

Actividad 4: Gestión de Mensajes

Objetivo: Permitir al usuario poder recibir y enviar mensajes.

Historia de Usuario:

Historia 4.1: Enviar mensaje

-Como usuario, quiero ser capaz de enviar un mensaje de texto o emoticono a un nombre de contacto, grupo o número de teléfono.

-Criterios de Aceptación: Puedo ser capaz de enviar un mensaje si el número de teléfono existe o si el nombre del contacto o grupo está registrado.

Historia 4.2: Recibir mensaje

-Como usuario, quiero poder recibir mensajes de mis contactos o de un número de teléfono que no tenga registrado.

-Criterios de Aceptación: Puedo ser capaz de recibir mensajes si estoy registrado en la aplicación.

Actividad 5: Búsqueda de mensajes

Objetivo: Facilitar la búsqueda de mensajes en la aplicación a través de diversos criterios de búsqueda.

Historia de Usuario:

Historia 5.1: Buscar mensajes por texto, número de teléfono y por contacto

-Como usuario, quiero buscar mensajes por texto, número teléfono o por nombre del contacto para encontrar fácilmente los mensajes que me interesan.

-Criterios de Aceptación: Debo poder escribir un texto o subcadena en los campos de búsqueda para el texto, un número de teléfono que identifique al emisor o receptor, o un nombre de contacto.

Actividad 6: Funcionalidad Premium

Objetivo: Ser capaz de exportar en un documento PDF los mensajes intercambiados con otro usuario indicado.

Historia de Usuario:

Historia 6.1: Activar cuenta Premium

-Como usuario, quiero acceder a las funcionalidades que ofrece ser premium como puede ser el generar un archivo PDF de los mensajes.

-Criterios de Aceptación: La cuenta debe estar registrada en la aplicación y realizar el pago de manera satisfactoria y si no se completa el pago, saltara un mensaje de error. La aplicación permite aplicar posibles descuentos (mensajes o fecha).

Historia 6.2: Desactivar cuenta Premium

-Como usuario, quiero poder desactivar mi cuenta premium.

-Criterios de Aceptación: La aplicación desactiva la cuenta premium y se quita el acceso a las funciones adicionales.

Historia 6.3: Generar documento PDF

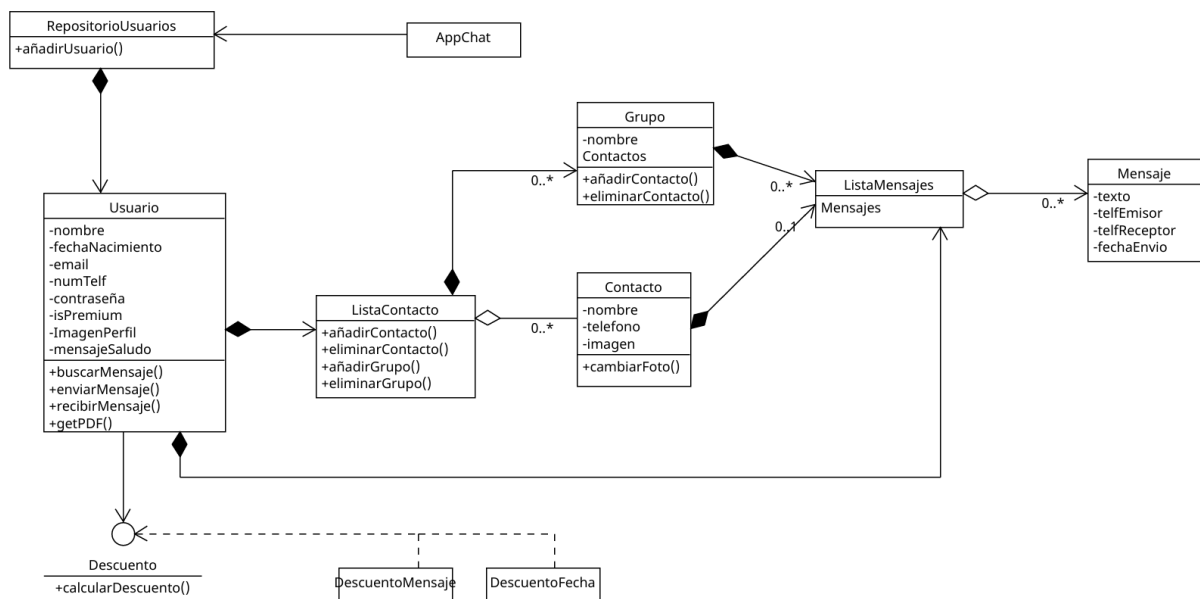
-Como usuario premium, quiero ser capaz de generar un archivo con mis contactos, mensajes, grupos y teléfonos.

-Criterios de Aceptación: La cuenta que haga la generación de PDF debe ser premium y si no lo es, le saldrá una opción de actualizar la cuenta a una premium.

Diagrama de Clases de Dominio:

La primera versión inicial de nuestro diagrama de clases es la siguiente, pero a medida que íbamos desarrollando nuestra app, tuvimos que hacer varias modificaciones, entre ellas destaca en tener solamente una clase contacto que implemente un contacto individual o un grupo de contactos.

Siendo la primera versión la siguiente;



Y la versión final del diagrama de clases, el siguiente esquema donde podemos observar que se asemeja bastante al esquema inicial.

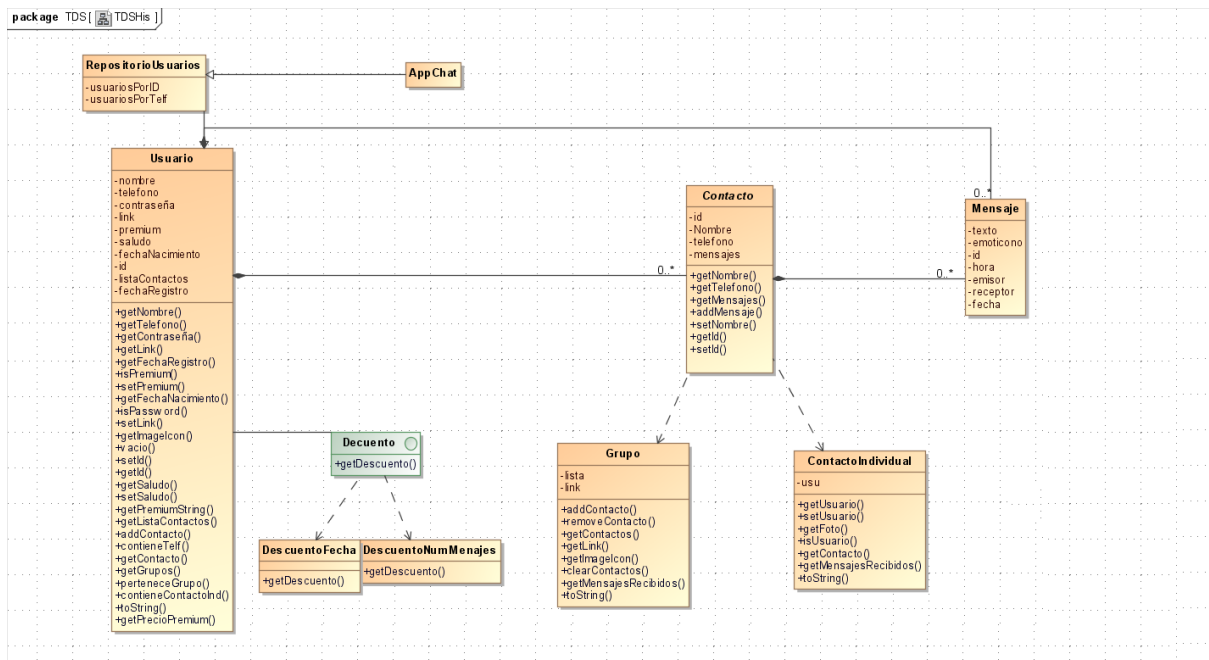
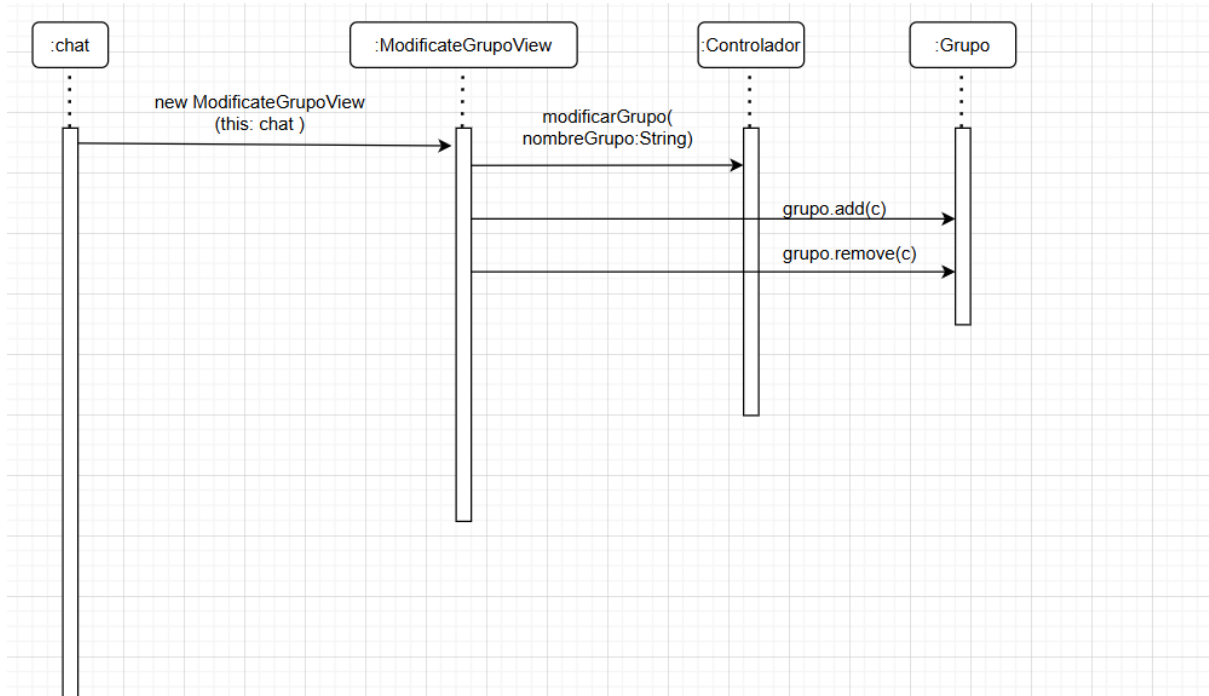


Diagrama de Operación de Añadir Contacto a Grupo:

En la siguiente gráfica podemos ver el diagrama de comunicación que se realiza para la operación de añadir/suprimir un contacto a un grupo existente.



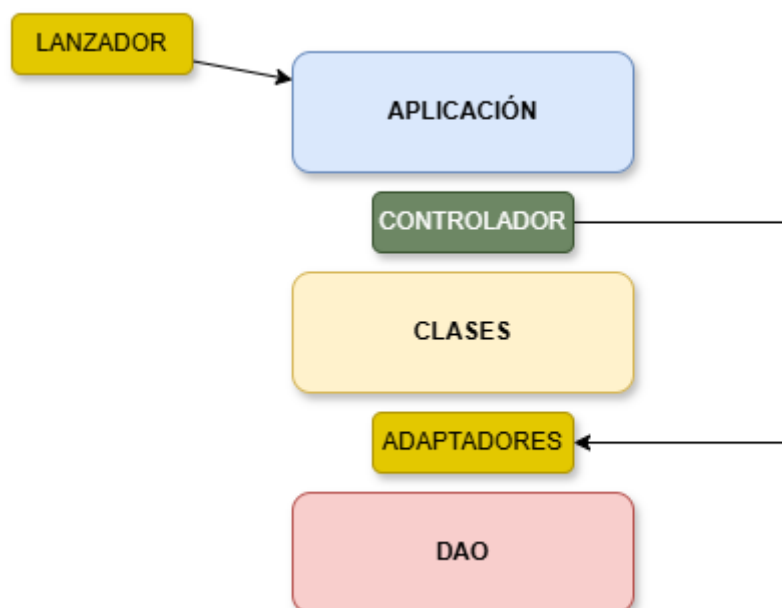
En nuestro proyecto, para modificar un grupo ya existente y ser capaz de añadir o eliminar contactos, el usuario puede hacerlo de manera dinámica en la ventana “ModificateGrupoView” donde es necesario escoger un grupo previamente, tras ello se puede ir seleccionando y eliminando o añadiendo los contactos agregados al grupo.

Estas acciones ya van modificando el objeto Grupo, pero hasta que no se confirma con el botón Confirmar, estos cambios no se registraran en la Base de Datos.

Cuando se presiona Confirmar, el controlador se encarga de actualizar la entidad Grupo correspondiente. Esto incluye la sincronización automática de la lista de contactos asociada en la base de datos, asegurando que la base de datos quede alineada con los cambios realizados por el usuario.

Arquitectura de la App:

Nuestra aplicación se organiza en cuatro capas semejantes a las estudiadas en teoría. La capa de Aplicación (presentación) que consta de 18 clases encargada del modelaje de las vistas, la capa de Clases (dominio) encargada de las clases del dominio y los repositorios, la capa DAO (servicios técnicos) encargada de la persistencia de las distintas entidades presentes del programa y los servicios técnicos. Además, existe un controlador que actúa como fachada entre las vistas y el dominio garantizando el principio de separación Modelo-Vista. Y, por último, una capa Lanzador encargada de ejecutar la aplicación. Un breve esquema sería el siguiente:



Como podemos observar la clase lanzador es la encargada de lanzar la aplicación comunicándose con la capa de aplicación concretamente con la ventana de registro, a continuación, el controlador se comporta como una fachada garantizando la separación la lógica entre las vistas y la lógica de negocio, también se comunica con la capa de persistencia para actualizar la información en la base de datos de cualquier operación relacionada con las clases Usuario, Contacto Individual, Grupo y Mensaje. Algunos aspectos interesantes son la existencia de una carpeta resource que contiene todas las imágenes que se van a utilizar en el proyecto.

Decisiones de interés para comprensión:

A lo largo del desarrollo del proyecto, nos enfrentamos a diversas decisiones para garantizar que nuestra aplicación se asemejara lo máximo posible a una aplicación de mensajería real. Algunas de las decisiones clave que tomamos fueron las siguientes:

- Durante el registro, se requiere que el usuario complete un campo de apellidos y confirme la contraseña para evitar errores de inicio de sesión causados por posibles erratas. Además, tan solo se permite imágenes con formato png para menor complejidad.
- Si un grupo o usuario no cuenta con una foto de perfil o esta falla al cargarse, automáticamente se le asigna una imagen predeterminada.
- A la hora de codificar las clases, estuvimos debatiendo sobre si era mejor tener dos clases separadas "Contacto" y "Grupo" pero creíamos que era más conveniente tener una clase padre debido a que ambas clases tenían una funcionalidad similar, estas clases son "Grupo" y "Contacto Individual" y la clase padre es "Contacto"
- Para permitir la recepción y respuesta de mensajes de usuarios desconocidos, implementamos los llamados "contactos fantasmas". Cuando un usuario recibe un mensaje de alguien que no tiene agregado, automáticamente se crea un contacto temporal sin nombre y con solo presencia de su número de teléfono. Este contacto puede ser editado posteriormente para añadir un nombre si el usuario decide agregarlo.
- Una forma de representar que un usuario es premium es cambiar el color de su nombre en el perfil a uno que sobresalga y destacando su estado y acceso a funcionalidades exclusivas como la exportación de PDFs.
- A la hora de buscar mensajes por nombres, si existe un contacto que no esté agregado en la casilla de Emisor o Receptor saldrá la cadena vacía, es necesario agregar el contacto si queremos saber quién es el emisor o receptor.
- Para modificar un contacto o grupo, este debe estar previamente seleccionado. Esta decisión se tomó para aprovechar el comportamiento natural de un modelo, donde la selección de un elemento se mantiene hasta que se elige otro.
- Los mensajes incluyen el nombre del contacto tal como ha sido agregado por el usuario. Si el emisor no está agregado, se muestra su nombre de usuario. Una vez agregado como contacto, el nombre del emisor se actualizará automáticamente en los mensajes. También, para controlar la bidireccional de los mensajes hemos utilizado una estructura pool semejante a la proporcionada en el proyecto de Tienda.
- A la hora de exportar un PDF, hemos decidido que solo sean los usuarios que tenemos agregados como contactos, no todos los contactos del repositorio.

- Otra decisión que hemos decidido que en vez de que salgan los nombres de los contactos en vez de los usuarios que están enlazados al contacto. Hemos tomado la decisión ya que, al ser grupos privados, los miembros solo los sabe el que ha creado el grupo y no el resto.
- Una de las decisiones de diseño más importantes fue determinar cómo afrontar la representación de un grupo. Finalmente, decidimos que un grupo sería una forma de distribuir mensajes a varios contactos de manera eficiente. En lugar de que el grupo almacene los mensajes, su función principal será **reenviarlos** a todos los contactos que pertenezcan al grupo. Este enfoque convierte al grupo en una herramienta práctica para implementar "chats de difusión", donde los mensajes no necesitan persistir en el grupo y simplemente se propagan a los destinatarios.

Patrones de Diseño Usados:

A largo del proyecto, hemos utilizado diversos patrones de diseño de forma directa o indirecta sin tener conocimiento de su aplicación (utilizados por Swing) algunos de ellos son los siguientes;

Método Factoría. Para crear objetos instanciando a subclases diferentes como, por ejemplo, para crear un elemento de la interfaz usamos Usuarios y Contactos.

Factoría Abstracta. Para crear familias de productos independientes se usa en el Servicio de persistencia para crear objetos diferentes como un contacto, mensaje, grupo, etc.

Patrón Singleton. Consiste en garantizar que una clase tenga una única instancia y un único acceso global. Se utiliza en el servicio de persistencia, la clase controlador y el repositorio de usuarios.

Patrón Adaptador. Para convertir la interfaz de una clase en otra que una clase cliente espera, es utilizada en el servicio de persistencia.

Patrón Fachada. Como hemos mencionado anteriormente sirve para proporcionar una única interfaz a un conjunto de clases. Es utilizada en la clase controlador.

Patrón Composite. Para manejar objetos simples y compuestos, utilizada en la clase contacto que maneja contactos Individuales y grupos de contactos individuales.

Patrón Observer. Consiste en garantizar el cambio de todas las dependencias de un objeto cuando el objeto cambia, es usado para la funcionalidad premium y actualizar una conversión.

Patrón Command. No es utilizado de forma directa, pero swing lo aplica para manejar eventos de los botones.

Patrón Iterador: El patrón Iterador se usa de manera indirecta a través del uso de colecciones de Java que proporcionan iteradores para recorrer elementos como por ejemplo en la clase Repositorio Usuarios, se usa este patrón para recorrer las listas sin exponer la estructura.

Manual de Usuario:

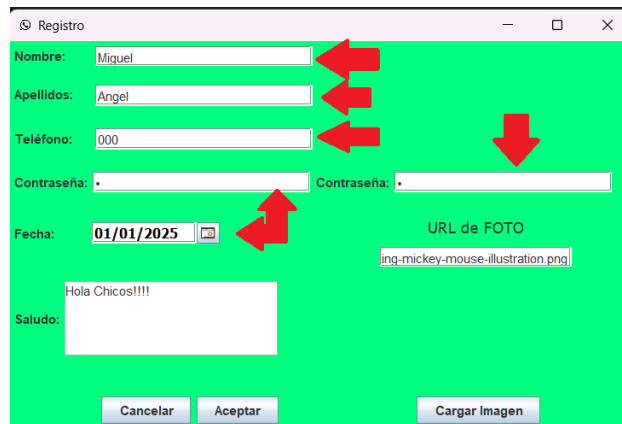
En este apartado, vamos a explicar los principales componentes de la aplicación, así como su uso y finalidad.

Registrarse como Usuario:

Nada más ejecutar la aplicación, nos saldrá la siguiente ventana que viene a ser la vista principal, donde se tendrá que pulsar en Registrar si no se tiene una cuenta ya registrada, en caso de tenerla se rellena los campos y se pulsa el botón aceptar.



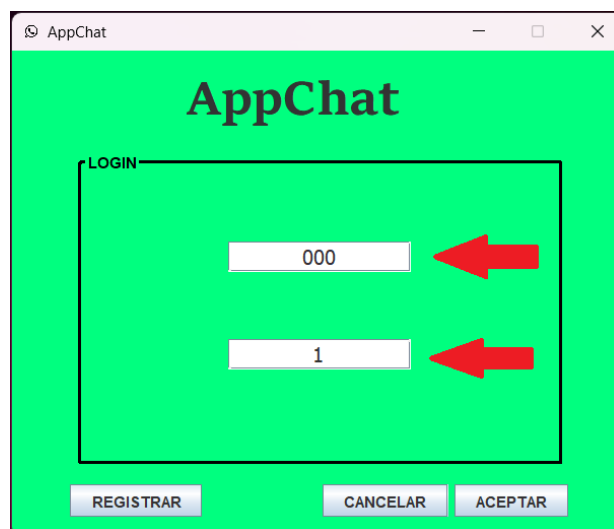
Si seleccionamos el botón de registrar nos saldrá la siguiente ventana que contiene los campos (señalados) donde es obligatorio rellenarlos para poder crear una cuenta, aquellos que no se rellenen, ya sea el saludo o la URL de la foto en internet (o cargar fotos del escritorio a través del botón) serán rellenados por defecto.



A screenshot of a web browser window titled "Registro". The form contains the following fields: "Nombre:" with the value "Miguel", "Apellidos:" with the value "Angel", "Teléfono:" with the value "000", "Contraseña:" (two fields), "Fecha:" with a date picker set to "01/01/2025", "URL de FOTO" with the value "img-mickey-mouse-illustration.png", and a "Saludo:" text area containing "Hola Chicos!!!!". At the bottom, there are three buttons: "Cancelar", "Aceptar", and "Cargar Imagen". Red arrows point to the "Nombre", "Apellidos", "Teléfono", "Contraseña" fields, the date picker, and the "Cargar Imagen" button.

Inicio de Sesión:

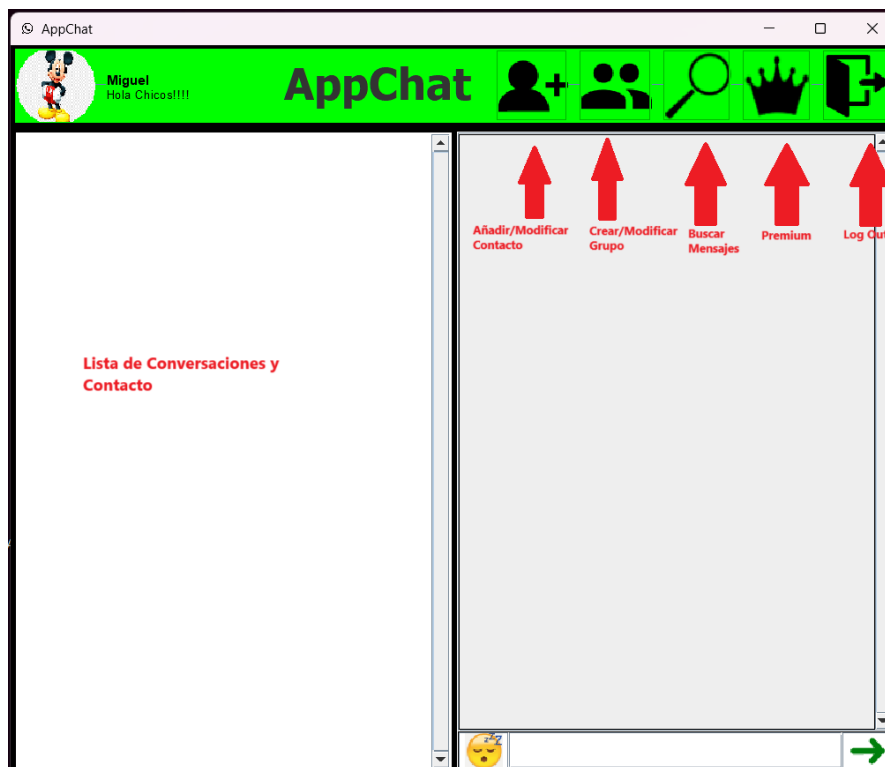
Tras registrarnos nos saldrá un mensaje de que de la operación se ha completado correctamente y nos redirigirá de vuelta a la pestaña de inicio de sesión, se rellenan los campos de teléfono y la contraseña. Y si todo sale bien, entraremos a la zona principal de la app.



A screenshot of a web browser window titled "AppChat". The main heading is "AppChat". Below it, there is a "LOGIN" section enclosed in a black border. Inside this section, there are two input fields: the top one contains "000" and the bottom one contains "1". Red arrows point to these two input fields. At the bottom of the window, there are three buttons: "REGISTRAR", "CANCELAR", and "ACEPTAR".

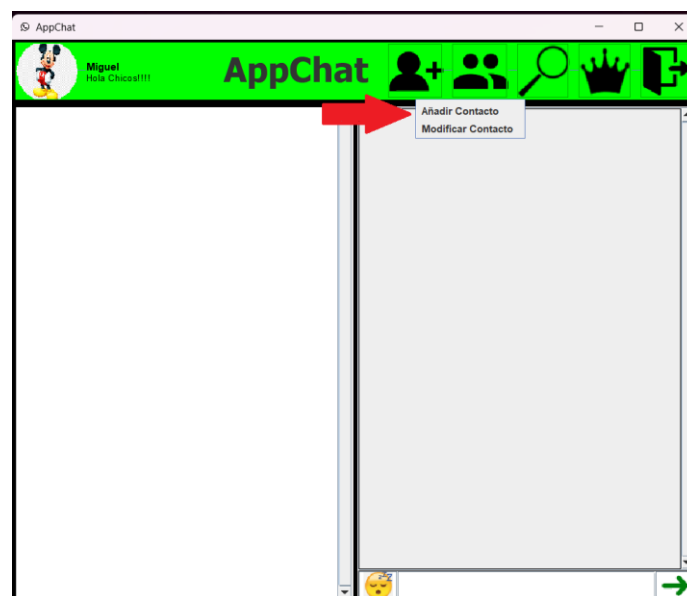
Ventana Principal de la Aplicación:

Al iniciar sesión, aparecerá la ventana principal con sus diferentes botones y funcionalidades que están disponibles. En la parte izquierda de la ventana estará la lista de contactos y la derecha la conversación con el contacto seleccionado.



Agregar/Modificar Contacto:

Dándole al primer botón contando desde la izquierda se nos abrirá un pequeño menú donde se nos da 2 opciones, modificar contacto y añadir contacto. Siendo el segundo el que abrirá una ventana donde poder rellenar información del contacto.



Donde se debe de introducir un número de teléfono válido (existente en la base de datos) y un nombre para ese contacto.



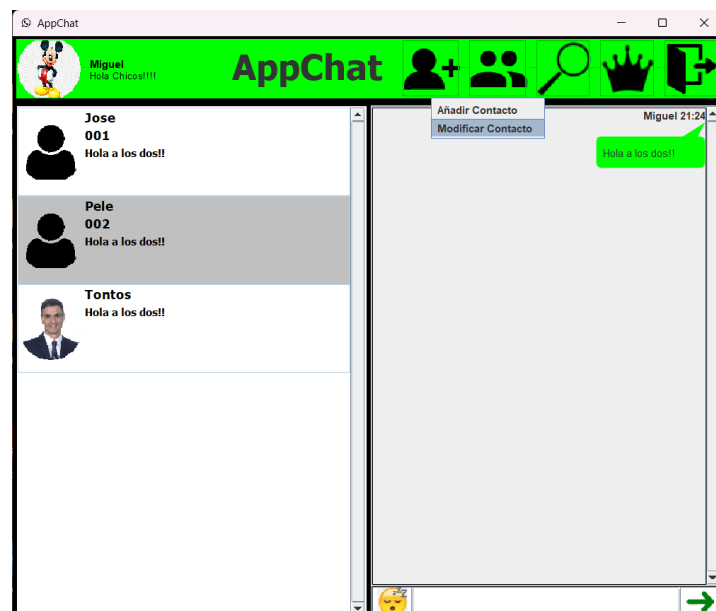
Añadir Contacto

Nombre: Jose

Teléfono: 001

ACEPTAR CANCELAR

En caso de querer modificar el contacto (hay que tener preseleccionado el contacto previamente) se elige la opción de Modificar Contacto y se hace un proceso parecido, donde solamente podremos cambiarle el nombre al contacto.



Como observamos, solamente se puede cambiarle el nombre a un contacto, es decir, al registrar un contacto con su número de teléfono este siempre existirá en la lista de contactos del usuario actual y, por tanto, no se podrá eliminar.



Modificar Contacto

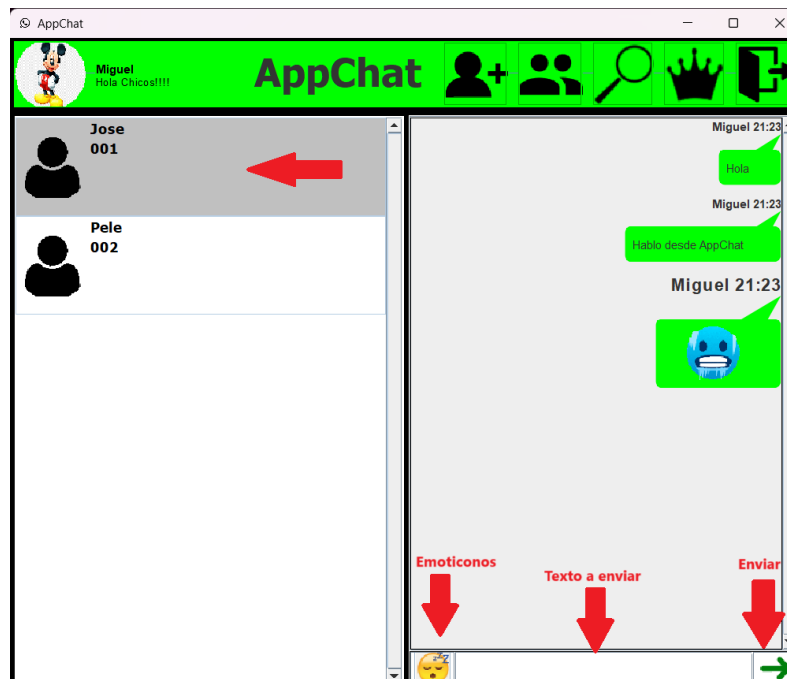
Nombre:

Teléfono: 000

ACEPTAR CANCELAR

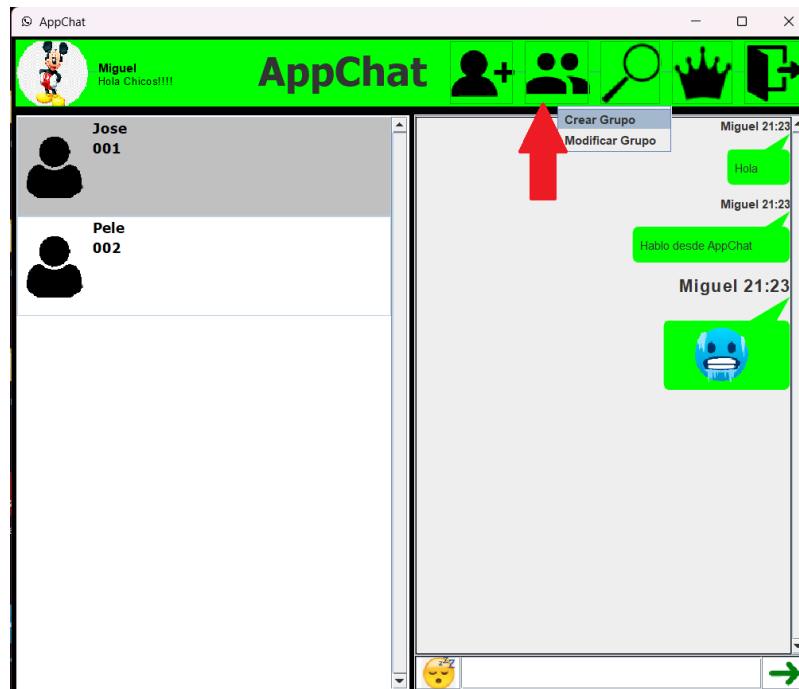
Abrir Conversación:

Una vez que tengas contactos creados, se abre el chat dándole 2 clics y con ello podemos mandar mensajes. Tenemos el botón de los emoticonos que abre una tabla y dándole al emoticono que queremos se mandara, para el texto hay que escribir en la caja de abajo y cuando se este seguro del mensaje, se le da a la flecha que hay ubicada a la derecha.

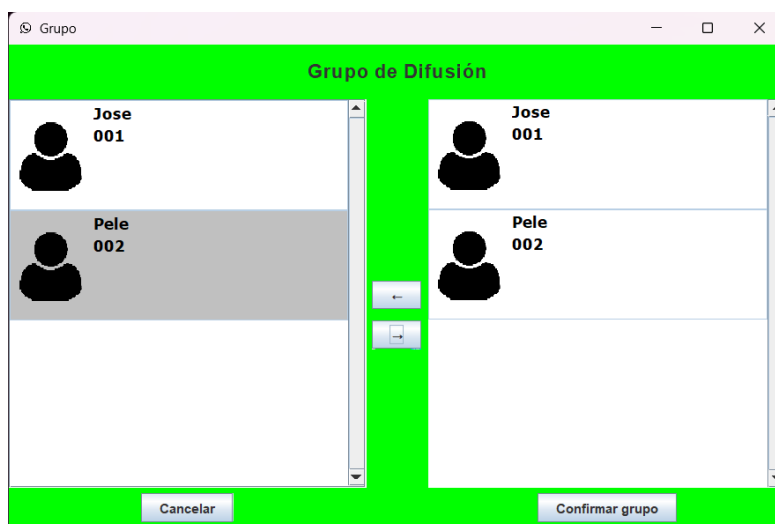


Crear/Modificar Grupo:

Para crear un grupo, apretamos el botón de los grupos, abre el menú y escogemos la opción que sería en este caso "Crear Grupo" que se abrirá otra ventana:



En esta ventana saldrán todos los contactos que tenemos agregados y para escoger que contactos queremos agregar, los seleccionamos y se le da a la flecha hacia derecha, si queremos rectificar y eliminar un contacto lo seleccionamos de la parte derecha y le damos a la flecha de la izquierda. Cuando eso esté terminado se le da a “Confirmar Grupo” y se abrirá otra ventana. En caso de salir querer insertar el mismo contacto dos veces al grupo saldrá una advertencia.



Tras confirmar el grupo que queremos crear, nos saldrá una ventana resumen que contendrá los nombres de los integrantes seleccionados anteriormente, un nombre para identificar el grupo y de forma opcional una foto para el grupo que se carga de forma local, en caso de no asignar nada existirá una por defecto. Tras rellenar todos esos campos y darle al botón confirmar ya tendremos el grupo creado correctamente.

Grupo

Resumen del Grupo

Nombre: Tontos

Integrantes:

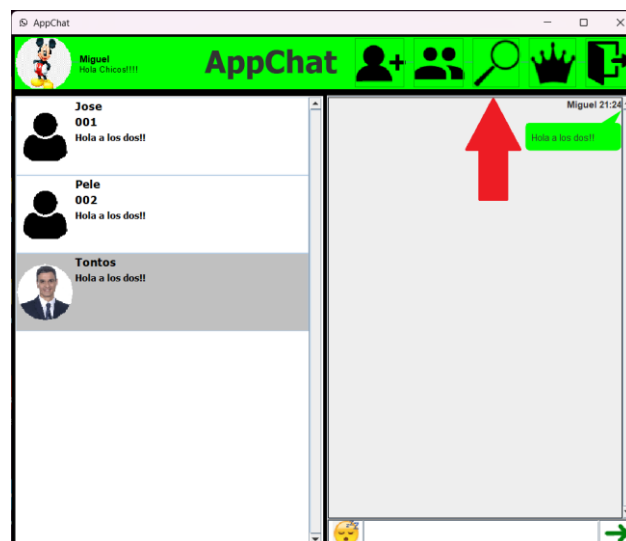
Jose
Pele

cargar

Confirmar Cancelar

Buscar Mensajes:

Para buscar mensajes que ha enviado o recibido el usuario, se seleccionara el botón de Buscar:



Y saldrá la siguiente ventana, donde podemos ver tres campos texto del mensaje para realizar la búsqueda, el número de teléfono que se deberá de especificar si es de tipo emisor (el que envía el mensaje) o de tipo receptor (quién recibe el mensaje) y, por último, el nombre del contacto que se debe especificar de qué tipo es igual que el número de teléfono. Una vez rellenado los filtros se debe de darle al botón buscar para realizar la búsqueda.

Buscador de Mensajes

buscar BÚSQUEDA

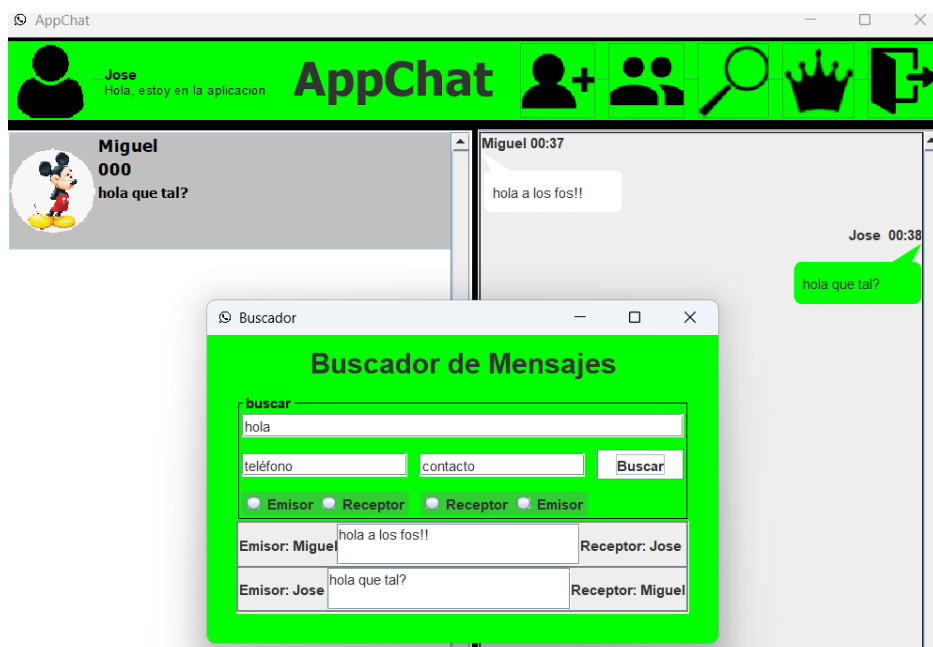
mensaje ← INDICAR EL MENSAJE

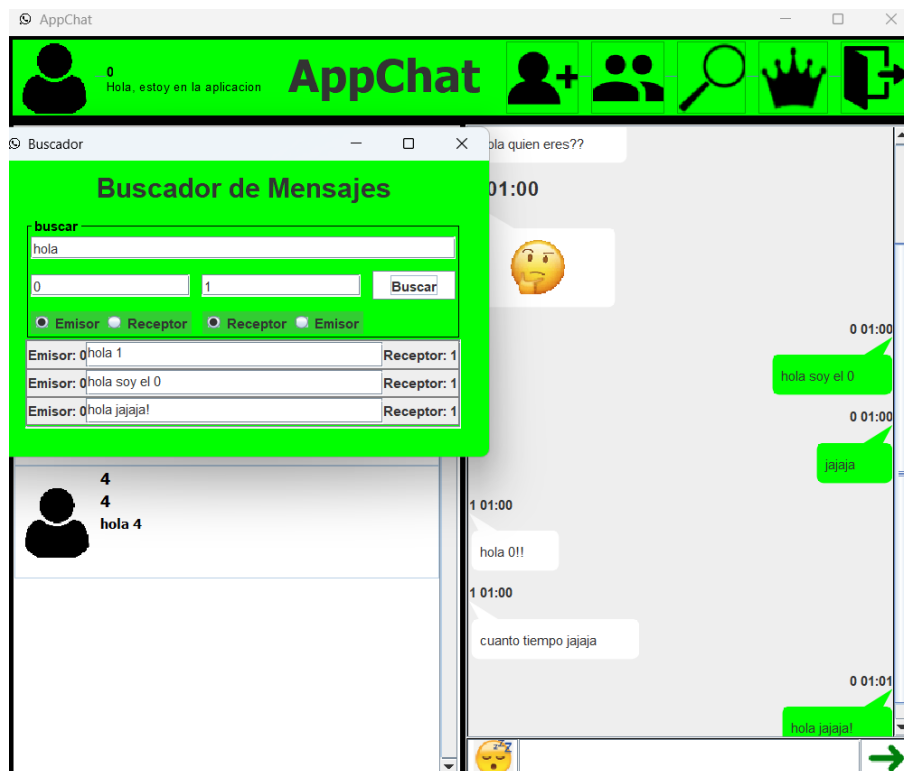
teléfono ← INDICAR EL TLF contacto ← INDICAR EL NOMBRE **Buscar**

☐ Emisor ☐ Receptor ☐ Receptor ☐ Emisor ← FILTROS

RESULTADOS DE LA BÚSQUEDA

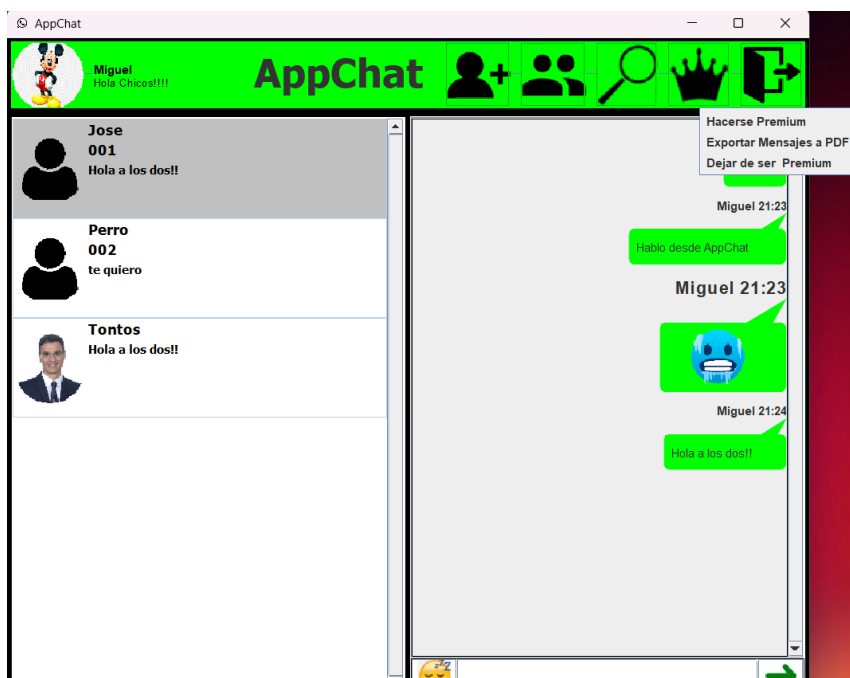
Para ver su funcionamiento, vamos a buscar un mensaje cuyo filtro sea la cadena hola en los siguientes ejemplos. Como podemos observar en la siguiente imagen la búsqueda con solo ese filtro nos muestra los dos únicos mensajes existentes para el usuario José. También podemos observar los resultados de la búsqueda en el usuario Pele.



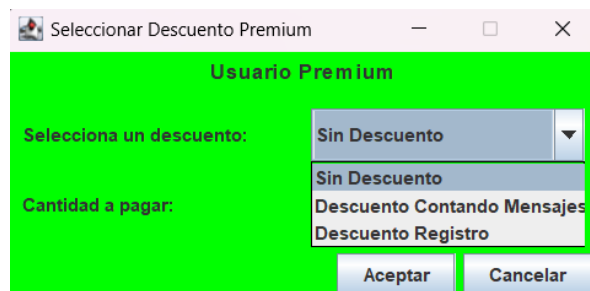


Funcionalidades Premium:

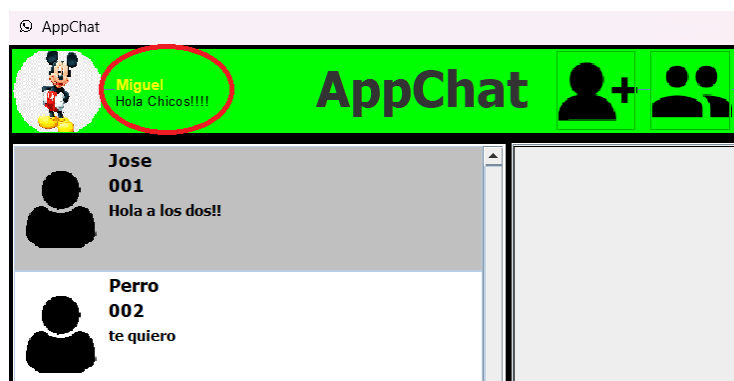
Hay ciertas funcionalidades que solo se pueden usar si el usuario actual es premium, para ello, hay que hacer clic en el botón de la corona y elegir la opción "Hacerse Premium":



En esta ventana, nos saldrá el coste de hacerse premium, pero se puede escoger el descuento que veamos más conveniente ya que al escoger un descuento, se actualizará el precio aplicando el descuento



Para comprobar si somos premium, tan solo tenemos que mirar si el nombre de nuestro usuario es de color amarillo, si es así es que nuestra cuenta se ha vuelto premium. Para renunciar a las funcionalidades de ser una cuenta premium tan solo debemos de elegir la opción de “Dejar de ser premium” y ya nuestra cuenta contendrá la funcionalidad por defecto.



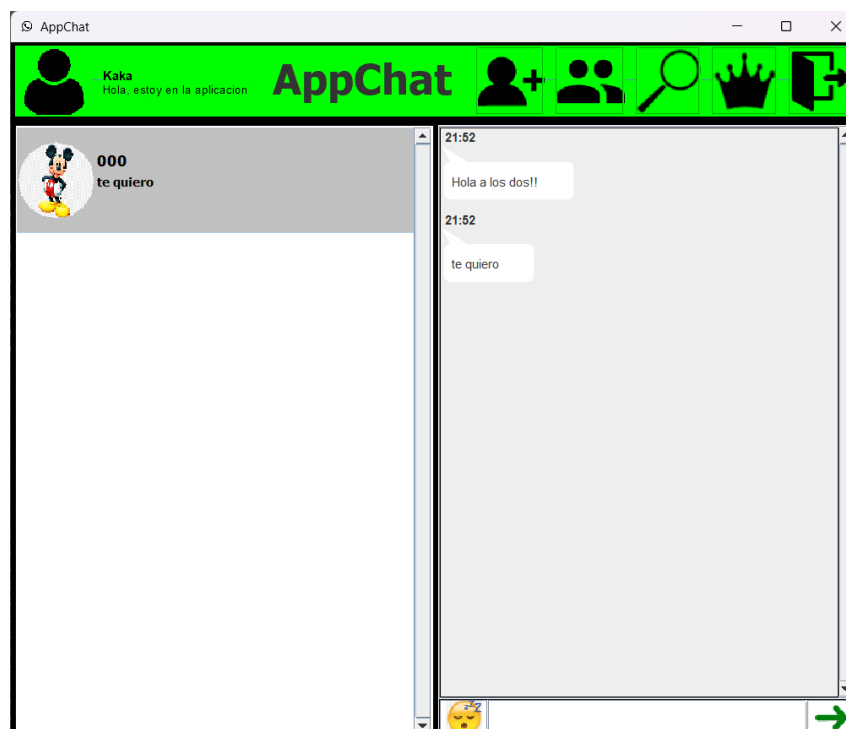
Con la cuenta premium, podemos ya usar los beneficios que conllevan, como la exportación de mensajes de los usuarios que tenemos agregados y a los grupos a los que pertenecen:

```
Usuarios de la aplicación:
Kaka   Telefono: 002
[]
Luis   Telefono: 001
[]
Miguel Telefono: 000
[Grupo: Tontos
Miembros: [Luis   Telf: 001 , Kaka   Telf: 002 ]]
```

En este caso, de los usuarios que tenemos agregados ni Kaka ni Luis pertenecen a ningún grupo por lo que sale el conjunto vacío, en cambio, Miguel pertenece a “Tontos” y nos dan sus miembros.

Recibir Mensajes:

Como se especifica en las historias de usuarios, un usuario puede recibir mensajes de contactos que no tiene agregados como se puede observar en la siguiente imagen. Desde esta perspectiva el usuario actual puede agregarlo o seguir sin especificar ningún nombre:



Como se puede ver, el programa es capaz de detectar los mensajes de usuarios que no se tienen agregados y enseñarlos, siendo esta su principal característica con WhatsApp.

Observación Final:

Esta práctica nos ha permitido afrontar un proyecto con características similares a los que podríamos enfrentar en el ámbito laboral. A pesar de las largas horas de trabajo y los desafíos que enfrentamos durante el desarrollo de la aplicación, ambos coincidimos en que este proyecto nos ha ayudado a mejorar como programadores y profesionales de la informática.

A la fecha de entrega, el proyecto cuenta con más de 200 commits, lo que refleja el nivel de compromiso y dedicación que hemos puesto en él. Que se refleja en el siguiente enlace;

[the-HaMo/appChat](#)

Estamos profundamente orgullosos del resultado obtenido y del funcionamiento correcto de una aplicación tan compleja como esta. Este proyecto es el fruto de nuestro esfuerzo y perseverancia, y solo podemos estar agradecidos por haber logrado completarlo a tiempo y con éxito.

| | Mohammed Amrou Labied Nasser | Sergio García García |
|----------------------|------------------------------|----------------------|
| Login y Registro | 12 | 20 |
| Gestión de Contactos | 20 | 18 |
| Gestión de Mensajes | 25 | 25 |
| Búsqueda Mensajes | 20 | 6 |
| Función Premium | 5 | 6 |
| Persistencia | 5 | 40 |
| Ventanas | 40 | 5 |
| Clases | 6 | 10 |

Mohammed, con un total de 133 horas, y Sergio, con 130 horas invertidas en el proyecto, hemos demostrado un gran esfuerzo y dedicación. Cada uno ha aportado al proyecto según sus fortalezas, logrando un verdadero trabajo en equipo.

Es evidente que hemos dividido las tareas de manera equilibrada, reflejando una distribución equitativa del trabajo y el compromiso necesario para completar un proyecto de esta envergadura.