

Proyecto # 1	Redes
Diego Franco	20240

### Funcionalidades realizadas:

```

Bienvenido al cliente de mensajería XMPP
¿Que deseas realizar?
(1) Crear un nuevo usuario
(2) Iniciar sesión con uno existente
(3) Salir

```

```

*****
Menú de opciones:
1) Mostrar todos los contactos y su estado.
2) Agregar un usuario a los contactos.
3) Mostrar detalles de contacto de un usuario.
4) Comunicación 1 a 1 con cualquier usuario/contacto (Chat con un usuario).
5) Participar en conversaciones grupales.
6) Definir mensaje de presencia.
7) Enviar un archivo.
8) Desconectarse.
9) Eliminar cuenta.
*****

Ingrese el número de la opción deseada:

```

- Iniciar sesión al servidor con una cuenta existente.
- Cerrar sesión con la cuenta iniciada.
- Crear una nueva cuenta en el servidor.
- Mostrar una lista de contactos junto con su estado de disponibilidad y mensaje de presentación.
- Agregar nuevos contactos a tu lista de contactos, además de auto aceptar las invitaciones recibidas.
- Mostrar la información de un contacto en específico.
- Iniciar conversaciones individuales.
- Crear grupos y mandar invitaciones.
- Unirse a un grupo existente.
- Enviar mensajes a un grupo.
- Actualizar tu mensaje de presencia.
- Compartir y recibir archivos de manera fluida dentro del chat.
- Eliminar tu cuenta de usuario si es necesario.
- Recibir notificaciones.

## **Dificultades:**

Dentro de las dificultades de realizar este chat fue en primer lugar el utilizar una librería de la que no conocía absolutamente nada, agregándole también que nos había recomendado que no se realizara en Python por lo mismo de que las librerías no eran muy conocidas, pero realmente quise adentrarme en el reto y ver que tan complicado era.

Dentro del desarrollo del chat una de las primeras dificultades que tuve fue conectarme con el servidor, desde un inicio traté de utilizar la librería sleekxmpp pero nunca logré hacer la conexión correcta hacia el servidor. Entonces para poder solucionarlo tuve que cambiar de librería a slixmpp con la cual ya logré realizar la conexión tanto de inicio de sesión como de crear una cuenta nueva.

Otra de las dificultades que tuve a la hora de estar desarrollando el chat fue el recibir notificaciones sin la necesidad de estar cortando el hilo del menú principal, esto porque en Python la concurrencia es un poco complicada. Pero la manera en la que lo logré solucionar fue implementar la librería asyncio con la cual se puede simular el uso de hilos para estar ejecutando varios procesos de manera paralela. De esta forma logré hacer que las notificaciones no interrumpieran el flujo las acciones del usuario.

Una de las dificultades que pude encontrar fue el codificado de los estados de los usuarios, esto porque la librería utilizada al obtener el roster de los contactos trae consigo la codificación del estado. El problema radicaba en que la codificación del estado disponible y no conectado eran similares. El de disponible es "" mientras que el de no disponible es un objeto vacío, esto Python lo reconocía de la misma manera y siempre para estos dos estados los confundía. La manera en la que pude corregir este problema es definir que todos los usuarios de un inicio están con un estado de no disponible para que si el roster trae un objeto no vacío se cambie al correspondiente.

Otra dificultad con la que me tope fue que con la creación de los grupos a veces si funcionaba y a veces no me dejaba crear grupo. O que por ejemplo había ocasiones en donde no me dejaba crear grupos, pero si unirme a uno existente. La verdad para esta dificultad no existe como tal una solución porque más que nada era de parte del servidor que no tenía los permisos adecuados para la creación de grupos. Pero lo único que yo tuve que agregar para poder hacer uso de los grupos es que dentro de las configuraciones del grupo nuevo tuve que activar la opción de allowinvites. Ya que con esta opción activa todas las veces que lo he probado me ha funcionado.

Por último, con el paso de notificaciones me tope con que al tener abierta la app de gajim no me llegaban los mensajes a la terminal en donde estaba realizando las pruebas, esto porque la app como que interceptaba los mensajes antes de que llegaran al otro lado. Además, como que tiene alguna configuración de prioridad la app que si tienes una cuenta en la terminal y la misma en la app las notificaciones la recibirán la app. Por lo tanto, para solucionar esto únicamente tuve que cerrar la app para que no pudiera interceptar y poderlos recibir en la terminal.

### **Lecciones aprendidas:**

De los aprendizajes más notorios puedo destacar los siguientes:

Una de las cosas más valiosas que saqué de esta experiencia fue aprender a enfrentarme a librerías completamente nuevas. Al principio, admito que estaba un poco perdido, especialmente porque me habían sugerido evitar Python debido a las librerías poco conocidas. Pero decidí asumir el reto y ver hasta dónde podía llegar.

Otra lección valiosa surgió mientras desarrollaba el chat. Necesitaba recibir notificaciones sin interrumpir la experiencia del usuario en el menú principal. En Python, la concurrencia puede ser complicada, pero descubrí una solución utilizando la librería 'asyncio'. Esta herramienta me permitió simular hilos y ejecutar procesos en paralelo.

Un valioso aprendizaje que obtuve de este proyecto fue la importancia del trabajo constante y cómo puede ayudar a aliviar la carga del proyecto. Desde el principio, me sentí abrumado por la magnitud del desafío que tenía por delante. La idea de desarrollar una aplicación de chat con una librería desconocida parecía abrumadora y me preocupaba que fuera demasiado para manejar.

Sin embargo, a medida que me adentre en el proyecto y me comprometí a trabajar de manera constante, empecé a notar que realmente no era tan imposible como lo veía desde el inicio.