

**Proyecto # 1**  
**Redes**  
**José Mariano Reyes**  
**20074**

**Funcionalidades,  
dificultades,  
soluciones y lecciones  
aprendidas:**

# Funcionalidades

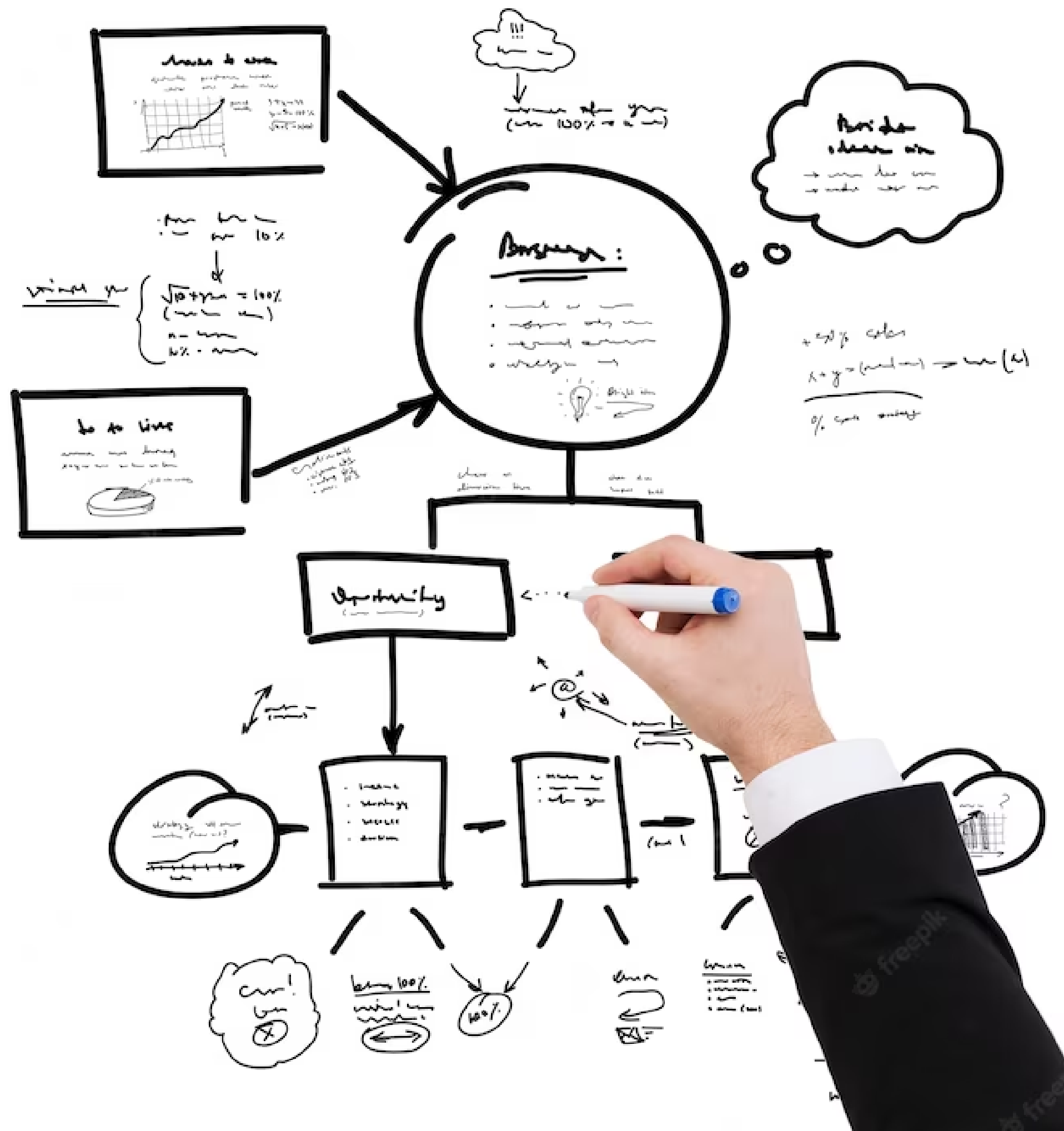
Se lograron implementar todas las funcionalidades propuestas en el documento.

Administración de cuentas:

- 1) Registrar una nueva cuenta en el servidor
- 2) Iniciar sesión con una cuenta
- 3) Cerrar sesión con una cuenta
- 4) Eliminar la cuenta del servidor

Comunicación:

- 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
- 5) Participar en conversaciones grupales
- 6) Definir mensaje de presencia
- 7) Enviar/recibir notificaciones
- 8) Enviar/recibir archivos



# Dificultades

Dentro de las mayores dificultades encontradas se encuentran:

1. Iniciar como tal el proyecto fue bastante difícil ya que mucho de lo que se probaba terminaba no funcionando al 100% o no funcionaba en absoluto.
2. El manejo de clientes asíncronos entre sí que lograsen interactuar en tiempo real.
3. Para los puntos extras fue bastante difícil implementar una librería en python que funcionara correctamente, ya que por alguna razón al implementar tkinter o similares los clientes se congelaban, como si un proceso siguiera ocurriendo en segundo plano.





# Soluciones

Para solucionar dichas dificultades se realizó:

1. ChatGPT me otorgó una planificación de todo el proyecto, donde me recomendó separar el código en 2 secciones principales y un archivo de manejo de menús. Al igual que en funciones específicas como enviar archivos fue bastante útil.
2. La biblioteca asyncio fue esencial para el desarrollo del proyecto ya que le daba un enfoque asíncrono a todo el proyecto para facilitar la comunicación de clientes.
3. Los puntos extras de GUI no se lograron conseguir por los problemas mencionados.







# Lecciones Aprendidas

1. Python, a pesar de ser posible realizar el proyecto, quizás no fue la mejor solución para implementar lo solicitado, más que todo por el manejo de clientes asíncronos con comunicación en vivo.
2. Existen librerías en Python que facilitan el proceso, entre ellas asyncio, xmpp y slxmpp destacan.
3. A pesar de haber logrado cumplir con tiempo las implementaciones solicitadas, hubiese sido mejor empezar un poco antes para conseguir los puntos extra de interfaz gráfica.
4. Seguir la estructura propuesta por ChatGPT fue una gran idea para darle un esqueleto al proyecto.

# Conclusiones

- XMPP a pesar de ser un protocolo ya establecido puede ser un poco difícil de implementar en ciertos lenguajes de programación.
- Las diferentes librerías que ofrece Python son esenciales para la programación del proyecto.
- Sin manejo de clientes asíncronos para tener mensajería en vivo, no es posible realizar el proyecto.
- Fue un proyecto entretenido de realizar con sus dificultades pero bastante útil.

# ¡Gracias!

Do you have any questions? [addyouremail@freepik.com](mailto:addyouremail@freepik.com)

+91 620 421 838

[yourcompany.com](#)

