

MANUAL DE USUARIO



BUAP

Materia: Sistemas Operativos I

Carrera: Lic. En Ciencias de la
computación.

Alumno: Osky

Profesor: Dr. Luis

Fecha: 06 de junio del 2020

Institución: BUAP

TABLA DE CONTENIDO.

INSTRUCCIONES. _____ ¡Error!
or! Marcador no definido.

Sobre la Aplicación. _____

¡Error! Marcador no definido.

Idea

Principal. _____ ¡Error!

Marcador no definido.

REQUERIMIENTOS. _____ 4

Requerimientos del sistema. _____ 4

Bibliotecas Requeridas. _____ 4

Bibliotecas en Windows. _____ 4

Bibliotecas en Linux. _____ 5

DISTINCION DE ARCHIVOS. _____ 6

Archivos del usuario. _____ 7

Archivos de audio. _____ 7

Archivos de imagen. _____ 7

Archivos del servidor. _____ 7

EJECUCION. _____ 8

Ejecución en Linux(servidor). _____ 8

Ejecución en Windows(servidor). _____ 8

Ejecución en Linux(cliente). _____ 8

Ejecución en Windows(cliente). _____ 9

DESCRIPCION. _____ 10

Contenido del menú de inicio. _____ 10

Contenido del usuario. _____ 11

USO. _____ 12

Del cliente udo/tcp _____ 12

Servidor _____ 13

INTRODUCCION.

El presente instructivo detalla los requerimientos y las configuraciones necesarias que el usuario deberá implementar en su herramienta de trabajo para la utilización del programa.

“Cabe mencionar que este programa es multiplataforma, funciona tanto en Windows como en Linux y la comunicación puede ser bilateral.”

Sobre La Aplicación.

Esta fue realizada bajo el lenguaje de programación Python2, debido a las nuevas demandas que implicaba la evolución del lenguaje se decidió migrarlo a Python3 esto para mayor compatibilidad futura y accesibilidad.

Idea Principal.

Lejos de lograr la comunicación entre cliente servidor se buscaba conocer y utilizar los distintos recursos que el lenguaje de programación Python3 ofrecía en este rubro. Logrando a si el tener comunicación punto a punto (cliente1 con cliente3 o cliente 4 etc.).

REQUERIMIENTOS.

Requisitos del sistema.

CPU.	Intel Celeron 800 MHz (Intel Core 2 Duo 2 GHz recomendados).
RAM.	256 MB (2 GB recomendados).
Espacio disponible en disco	200 MB.
Sistema operativo.	Windows 10/Linux.

¹ Requisitos mínimos

Teniendo esto en consideración nuestro programa funcionara correctamente aun que se recomienda que se obtengan mejores recursos para una mejor optimización.

Bibliotecas Requeridas.

Debido a que el proyecto esta enriquecido con algunas funciones particulares es necesario y de vital importancia instalar las siguientes bibliotecas. Estas normalmente no vienen predefinidas por el lenguaje Python y se tienen que instalar manualmente. Es por eso que a continuación se mostraran los comandos para poder instalarlas y ejecutar sin problemas el programa.

Bibliotecas En Windows.

Si utilizamos este sistema operativo, nos estaremos apoyando de la distribución **Anaconda** y para ser más específicos en el ejecutable **Anaconda Prompt** (esto se puede ubicar en el buscador de windows) el cual ejecutaremos.

Tomaremos en cuenta que las herramientas necesitan actualización y para eso ejecutaremos lo siguiente `python -m pip install --upgrade pip`.

pip: es el instalador "oficial" actual. Desde 2014 (Python 3.4), el módulo pip se instala junto con Python.

Procedente a lo anterior actualizaremos Python que en la fecha actual 06/06/20 la versión más actual 3.8.3 (Si en un futuro cambia verificar el tipo de instalación) con el siguiente comando.

```
conda install python==3.8.3
```

Una vez terminado los procesos de actualización continuaremos con la adquisición de las bibliotecas correspondientes.

Pygame. Es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos en dos dimensiones de una manera sencilla.

El comando de instalacion es la siguiente

```
python -m pip install -U pygame
```

Algunas librerias no fue necesario instalarlas debido a que el mismo instalador de python vino con ellas. Actualizando el lenguaje como su instalador y la librería anteriormente mencionada estariamos listos para ejecutarlo.

Bibliotecas En Linux.

Es preferente tener el minimo conocimiento para trabajar sobre este sistema operativo ya que se maneja terminal y comandos por lo cual trataremos de ser lo mas explicitos para no dificultar la tarea de instalacion.

Si no se tiene instalado python con el siguiente comando se instalara.
“sudo apt-get install python3”.

Continuaremos con las siguientes librerias.

Tkinder: es un paquete multiplataforma estándar de python para crear interfaces gráficas de usuario (GUI).

Su comando para instalar es el siguiente “apt-get install python-tk”.

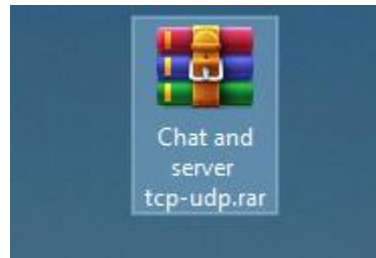
Pygame. Es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos en dos dimensiones de una manera sencilla.

Su comando para instalar es el siguiente “sudo apt-get install python3-pygame”.

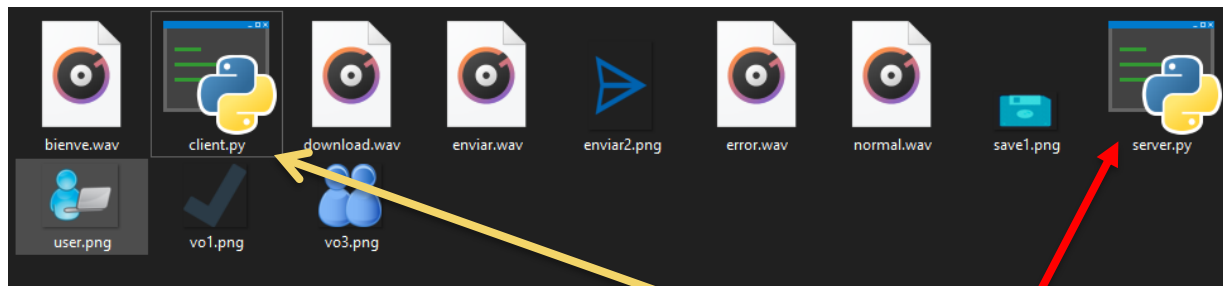
Habiendo seleccionado el sistema operativo de nuestra preferencia y realizando las instalaciones correspondientes procederemos a la descompresión del programa.

DISTINCION DE ARCHIVOS.

Entraremos en materia y procederemos a descomprimir el archivo en una carpeta sin ningún archivo y especialmente designada al chat.



Los archivos que contiene el rar son los siguientes:



Nos centraremos en los siguientes archivos de **client.py** y **server.py**

Client.py.- En este programa se encuentra el tipo de cliente que se quiere ser ya sea UTP o TCP con sus correspondientes interfaces

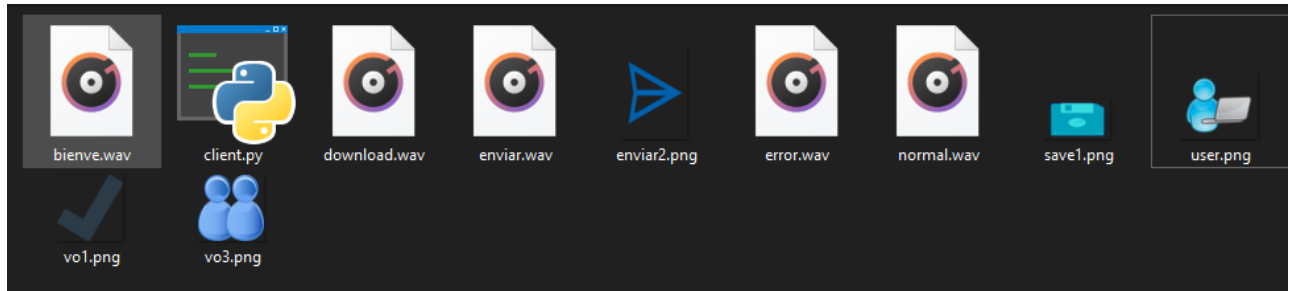
Server.py.- Aquí se aloja el código del servidor tanto para UDP como TCP el cual nos ayudara en la comunicación interprocesos.

Si se toma el papel de servidor solo se necesitara el código **server.py** puesto que no maneja interfaz y este sera el canal entre todos los clientes a si que se recomienda asignar una maquina especial para tal programa.

En dado caso que se tome el papel de cliente se tomaran todos los archivos exceptuando el código **server.py** y se guardaran en una carpeta especial. La cual se tendra que propagar sobre los distintos equipos en los cuales se quiere tener comunicación.

Archivos del usuario.

Para una mejor familiarización con lo que se maneja, se describe a continuación el contenido y el uso de cada archivo.



Archivos de audio.

- bienve.wav. -Sonido al mandar mensaje (tcp).
- download.wav. -Sonido al mostrar los usuarios(udp).
- enviar.wav. - Sonido al mandar mensaje en (udp).
- error.wav. - Sonido al no seleccionar el tipo de protocolo en el menú de inicio.
- normal.wav. -Sonido al mostrar usuarios (tcp).

Archivos de imagen.

- enviar2.png. - imagen del botón que manda el mensaje(udp).
- save1.png. - imagen del botón a la hora de registrar datos
- user.png. - imagen del botón a la hora de mostrar usuarios(udp).
- vo1.png. - imagen del botón que manda el mensaje(tcp).
- vo3.png. -imagen del botón a la hora de mostrar usuarios(tcp).

Archivos del Servidor.

Como se había mencionado antes el archivo del servidor no se le aplico interfaz, por lo cual no necesita de algún archivo extra.

EJECUCION.

Ejecución en Linux(servidor).

Como se había mencionado antes, es preferible tener conocimientos mínimos del sistema operativo para que sea más viable la ejecución.

Nos ubicamos en la carpeta del servidor y lo ejecutamos de la siguiente forma.

```
osky@osky:~/Escritorio/respaldo_2$ python3 server.py
Servidor conectado
```

Si lo hicimos correctamente nos mostrara la leyenda de **Servidor conectado** lo cual nos indicara que esta funcionando y a la espera de clientes que se conecten.

Ejecución en Windows (servidor).

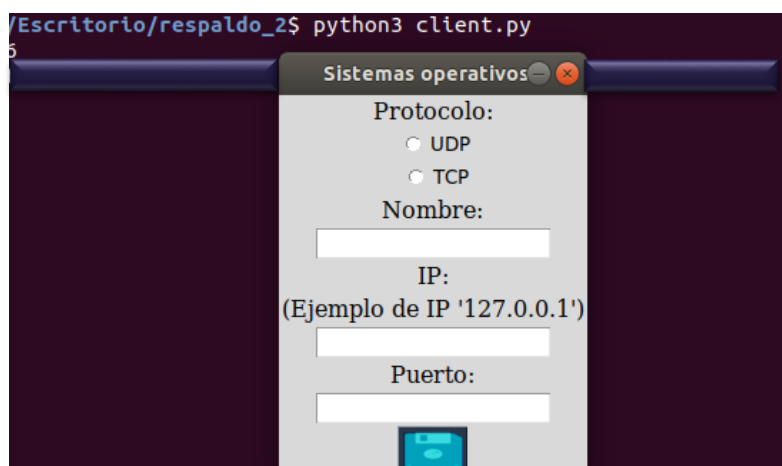
Retomaremos el **Anaconda Prompt** y nos posicionaremos en la carpeta donde se encuentra el archivo y ejecutamos lo siguiente.

```
(base) C:\respaldo_2>python server.py
Servidor conectado
```

Si no hay imprevistos imprimirá **Servidor conectado** lo cual nos indica que esta en espera de clientes.

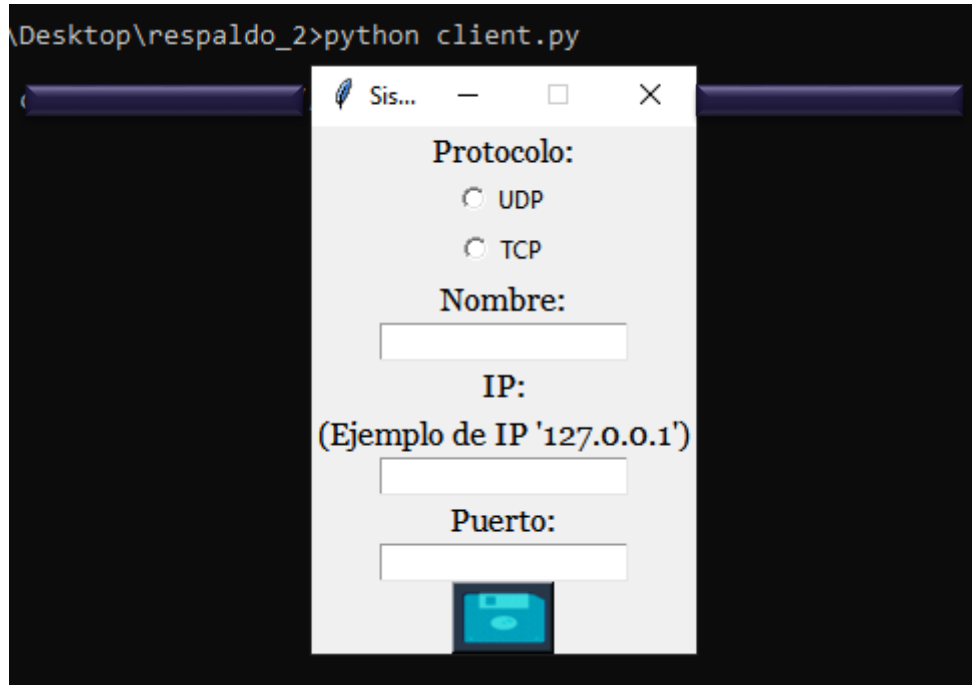
Ejecución en Linux(cliente).

Nos situaremos en la carpeta asignada para la ejecución del programa. Ejecutaremos lo siguiente y con éxito aparecerá una ventana denominada sistemas operativos.



Ejecución en Windows(cliente).

Apoyándonos de Anaconda Prompt ejecutamos lo siguiente y si fue con éxito aparecerá una nueva ventana

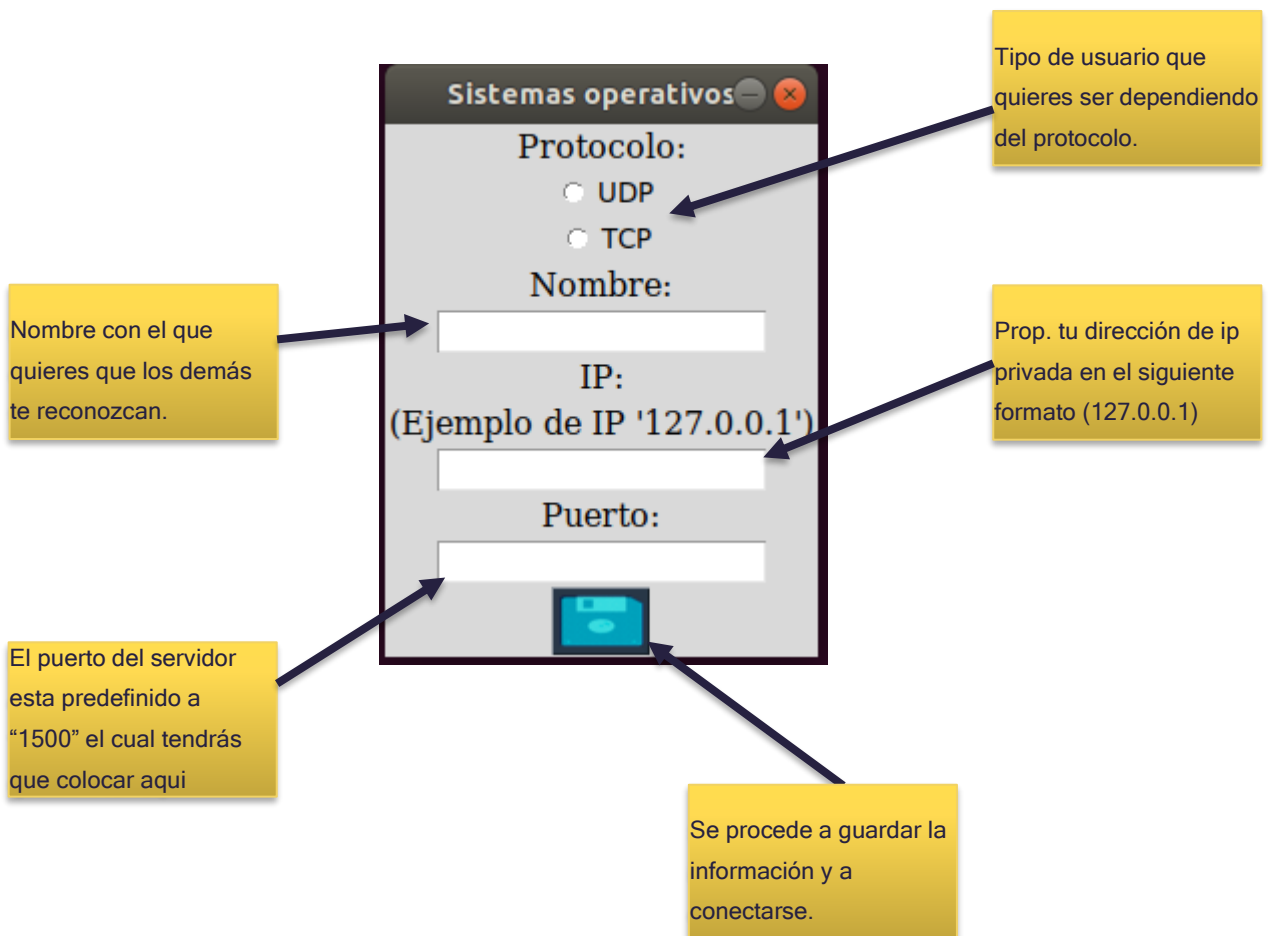


DESCRIPCION.

Contenido del menú de inicio. (cliente)

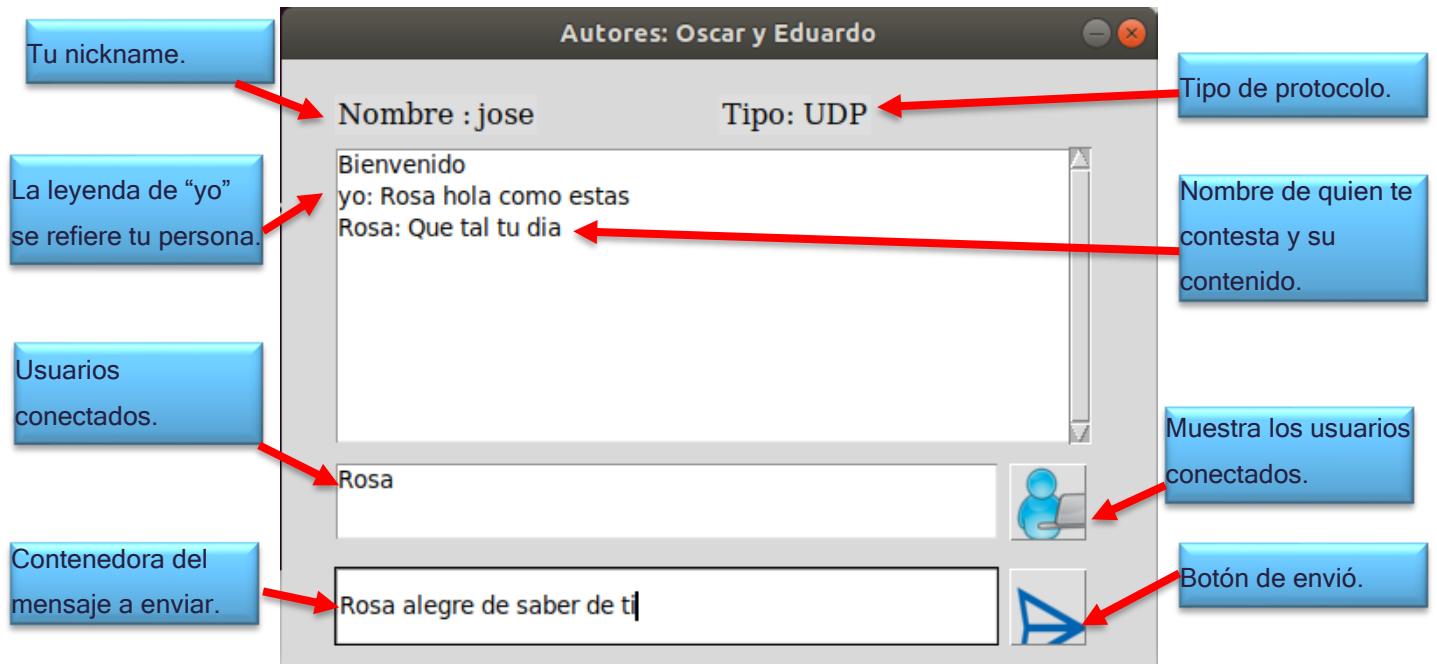
Para una mejor comprensión de la nueva ventana emergente se describirán los distintos apartados, así como su descripción.

Cabe resaltar que la interfaz tanto en Windows como Linux es muy semejante a si que englobaremos los 2.



Contenido del usuario.

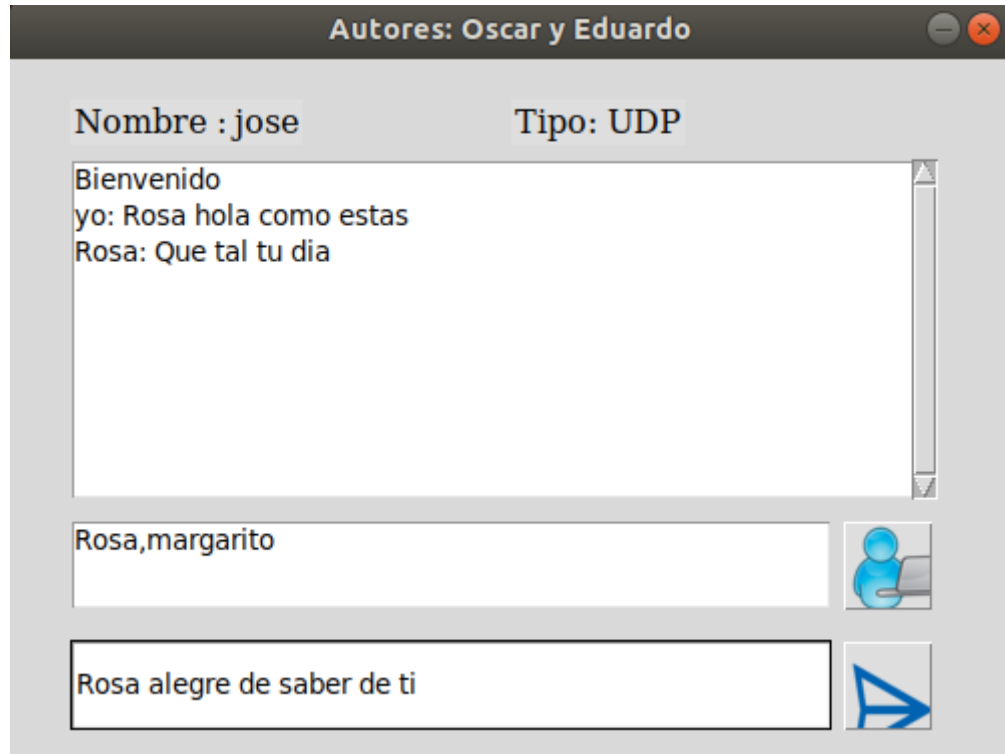
A continuación, indicaremos los distintos índices de la interfaz de usuarios.



USO.

Como tenemos la suficiente familiarización con el programa, el uso de este será más sencillo.

Tomaremos como ejemplo la ventana anterior y describiremos su uso

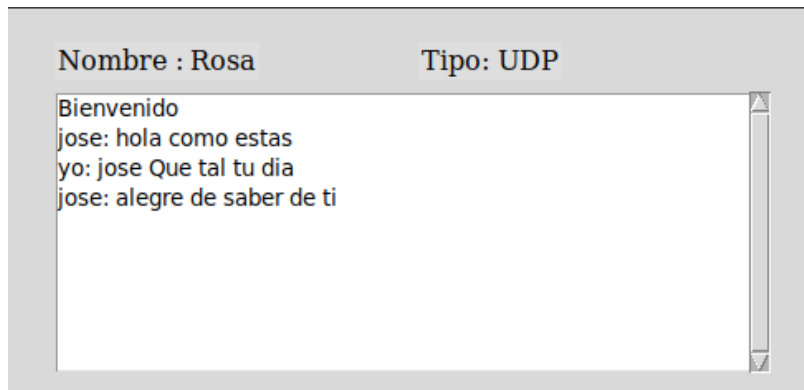


Para poder hacer uso de este es necesario que mínimo se tengan 2 usuarios conectados, además de que se utilice una mecánica en particular.

Carga de usuarios. - Para verificar que hay mas usuarios en línea presionaremos el avatar de color azul con el portatil. Esto nos imprimirá el nombre de todos los integrantes

Envío. - Para el envío de un mensaje a un destinatario en concreto. Se tendrá que definir al principio el nombre respetando mayúsculas a quien va dirigido el mensaje.

En este caso es "**Rosa**" procedente a este agregaremos el mensaje que le queramos enviar en este caso será "**alegre de saber de ti**" una vez terminado el mensaje presionamos el icono de 2 triángulos que simulan una carta en envio. Lo que hará que lo mande a travez del servidor hasta llegar al destino establecido.



Es así como el cliente Rosa recibe el mensaje y se muestra en su pantalla.

En la interfaz de tcp lo único que cambiaría es la interfaz de usuario puesto que la mecánica es igual.

Servidor.

Para una óptima monitorización en el servidor se mostrará el tipo de información canalizada, tanto los registros de usuarios como los mensajes que se mandan y las veces que cargan usuarios, como método de privacidad en este caso omitimos la impresión de mensajes.

```
servidor conectado
Informacio recibida UDP: registrar jose
Informacio recibida UDP: lista jose
Informacio recibida UDP: registrar Rosa
Informacio recibida UDP: lista Rosa
Informacio recibida UDP: lista jose
Informacio recibida UDP: mensaje Rosa
Informacio recibida UDP: mensaje jose
Informacio recibida UDP: registrar margarito
Informacio recibida UDP: lista jose
Informacio recibida UDP: lista margarito
Informacio recibida UDP: mensaje Rosa
```