

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD INTEGRAL DEL CHACO



“APLICACIÓN DE CHAT CLIENTE-SERVIDOR”

ESTUDIANTE:

EDSON NEIL CHILIMANI CUELLAR

MATERIA:

PROGRAMACIÓN DE APLICACIONES DE TIEMPO REAL – ELC-104

CAMIRI 29 DE JULIO DE 2023

Tabla de contenido

1	INTRODUCCIÓN.....	3
2	Requerimientos para utilización de la aplicación	4
3	Funcionamiento de la aplicación	5
3.1	Ejecutar el Servidor.....	5
3.2	Ejecutar el cliente.....	6
4	Envío de mensajes	7
5	Guardar mensajes	8
6	Envío de archivos	8

1 INTRODUCCIÓN

Bienvenido al manual de usuario de la aplicación de chat Cliente-Servidor. Este manual ha sido diseñado para brindar una guía completa sobre cómo utilizar la aplicación.



El proyecto fue creado en el lenguaje de java utilizando el IDE NetBeans, trata de la comunicación del cliente con el servidor utilizando socket, la aplicación cuenta con interfaz tanto en el cliente como en el servidor que son intuitivas y fácil de usar.

El aplicativo se hizo utilizando el sistema operativo “Windows”, solo se necesita tener instalado java-NetBeans sin importar la versión, ya que la aplicación se probó en diferentes versiones y funcionó con normalidad.

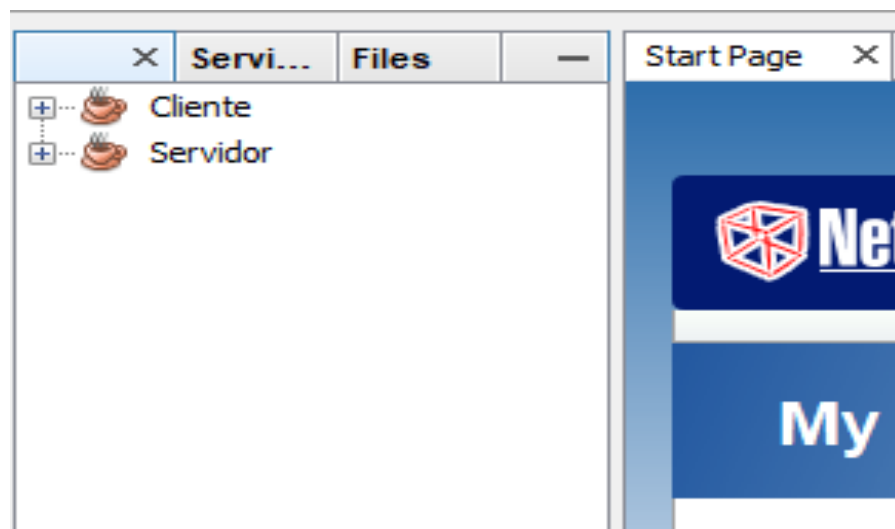
Podrá enviar mensajes y archivos, estos mensajes y archivos se mostrarán en un área de texto con la hora del envío, podrá guardar los mensajes en algún directorio de su computador que usted mismo especifique y los archivos enviados se guardarán en el directorio que se especifica en los códigos de la aplicación que mas adelante se mostrará para una mayor comprensión.

2 Requerimientos para utilización de la aplicación

Para poder ejecutar y probar el funcionamiento de la aplicación debe tener instalado java-NetBeans sin importar la versión, tener los 2 archivos tanto del cliente como del servidor.

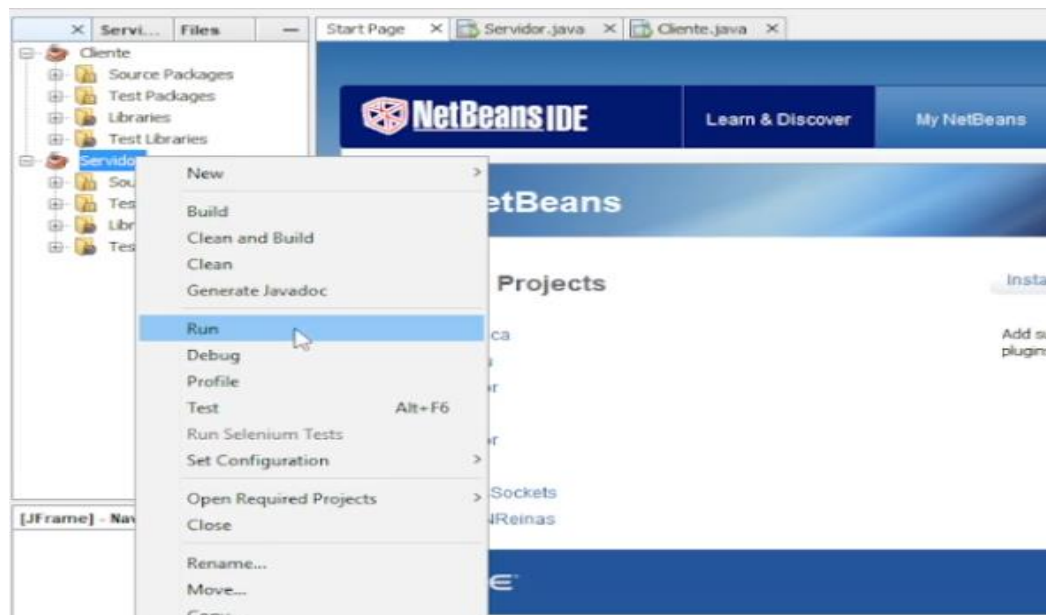
 Cliente	27/7/2023 14:05	Carpeta de archivos
 Servidor	27/7/2023 14:05	Carpeta de archivos

Estos dos archivos se deben abrir desde NetBeans

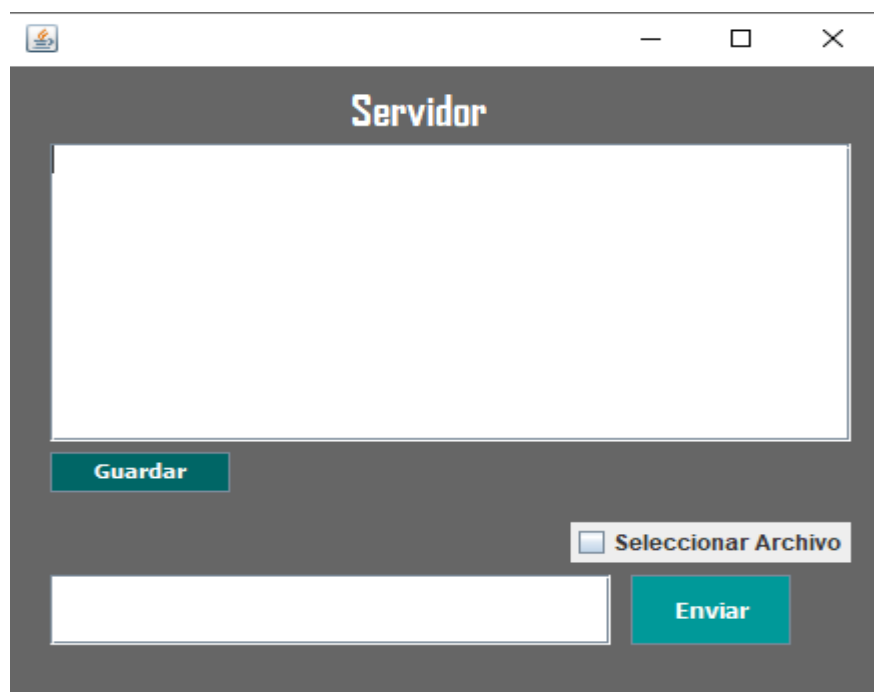


3 Funcionamiento de la aplicación

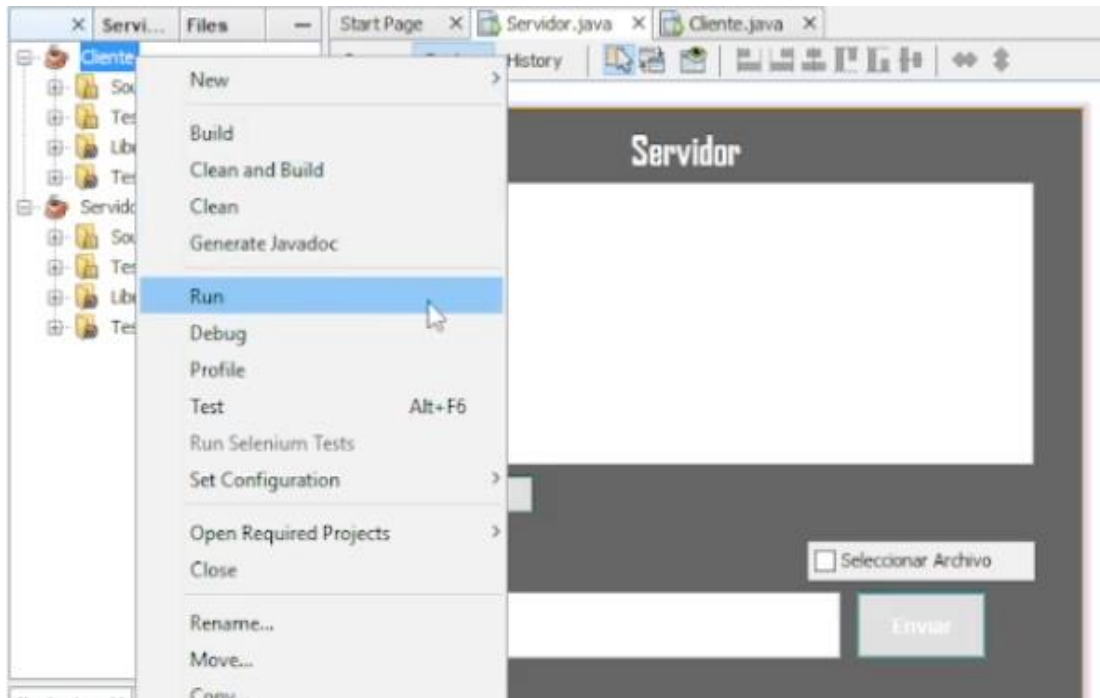
3.1 Ejecutar el Servidor



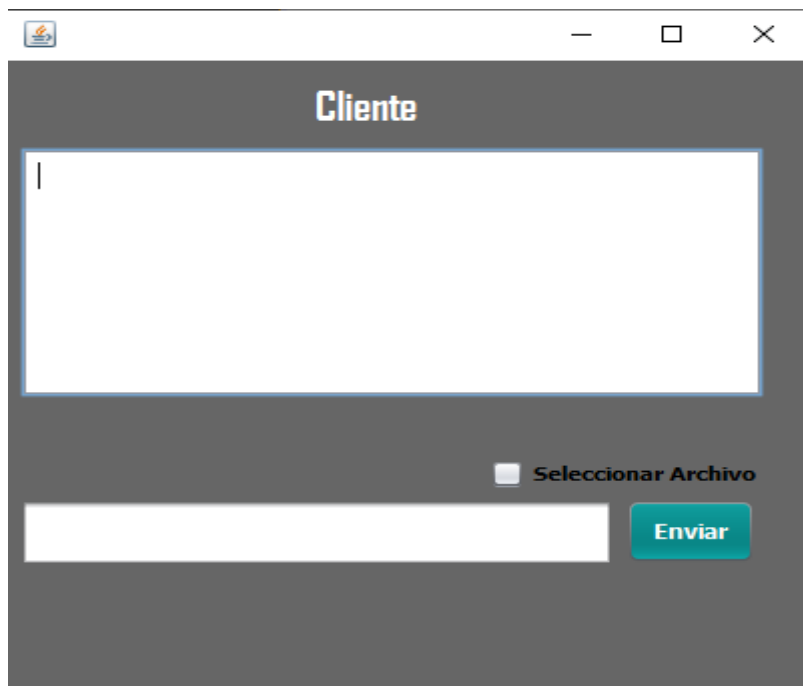
Al ejecutar el servidor se abrirá la siguiente ventana (ventana servidor)



3.2 Ejecutar el cliente

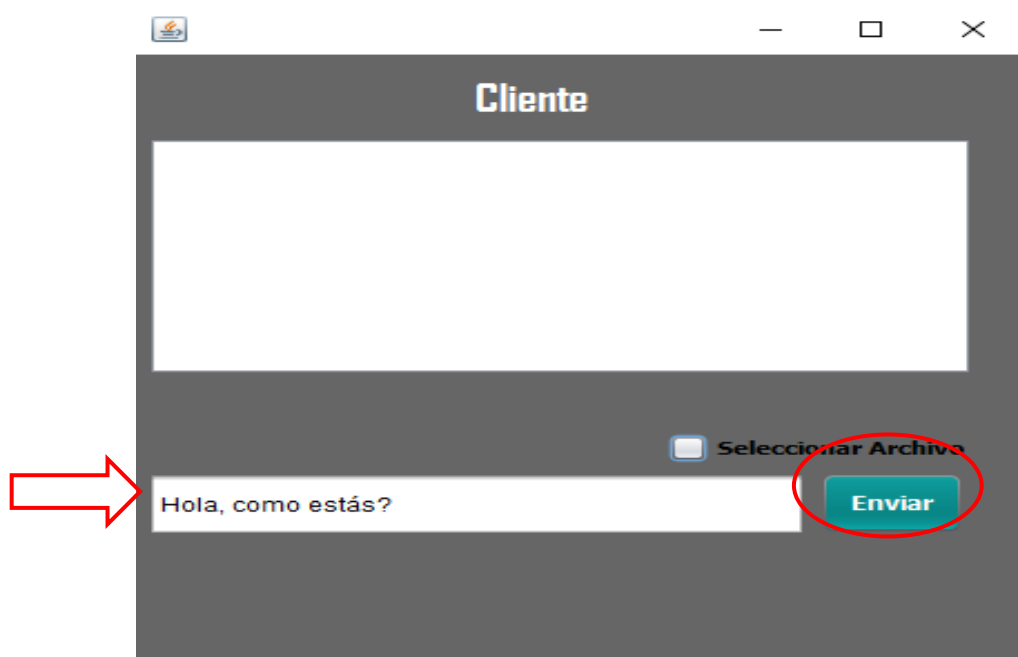


Al ejecutar el cliente se abrirá la siguiente ventana (ventana cliente)



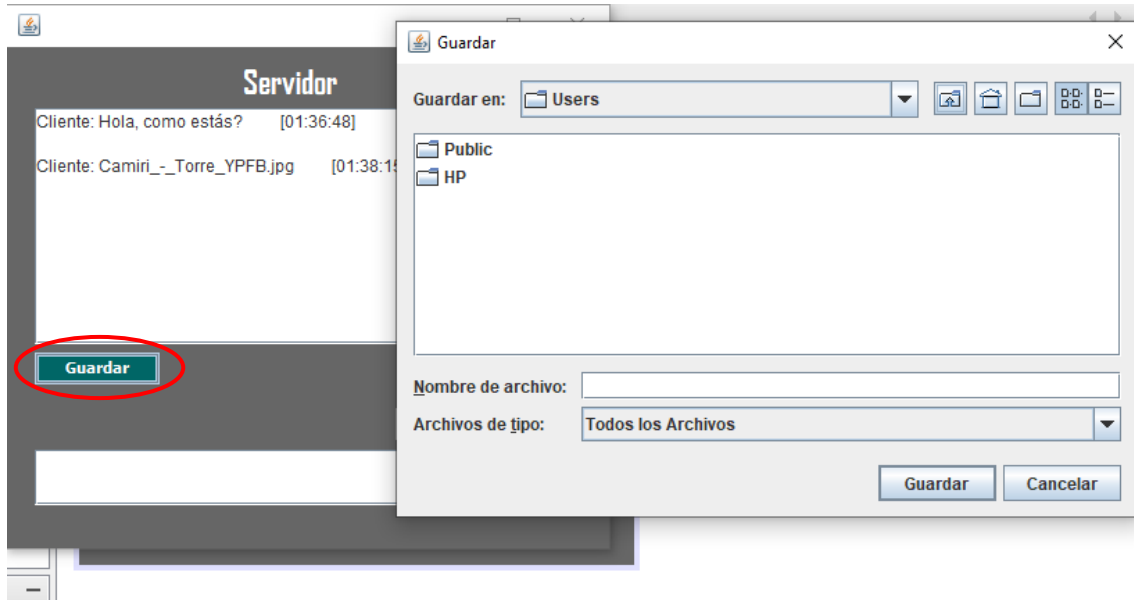
4 Envío de mensajes

Para poder enviar un mensaje se debe escribir el mensaje en el campo de texto, luego hacer clic en el botón “Enviar”, el mensaje se enviará y se mostrará en el área de texto del servidor si se está mandando el mensaje desde el cliente.



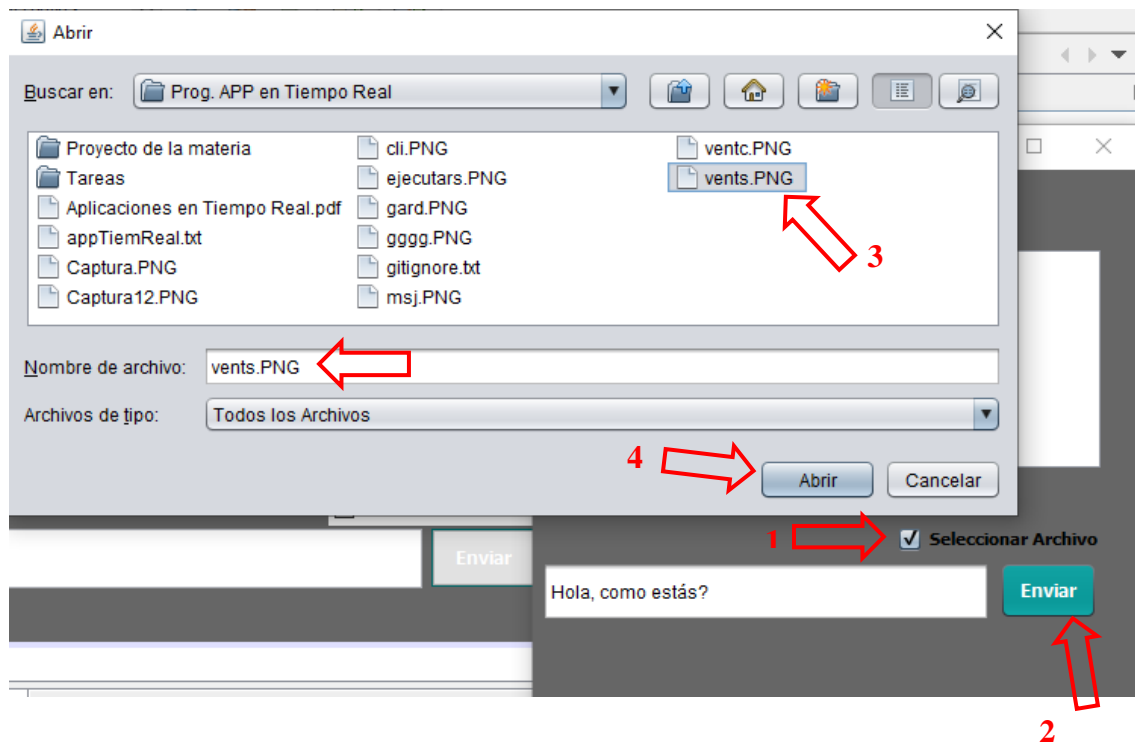
5 Guardar mensajes

Para guardar los mensajes se debe presionar el botón guardar, luego elegir el lugar donde se quiere guardar. Los mensajes se guardarán en el directorio especificado en un archivo de texto.



6 Envío de archivos

Para poder enviar archivos se tiene que marcar el botón “seleccionar archivo”, una vez marcado hacemos clic en el botón “Enviar” y se nos abrirá una ventana para que podamos seleccionar el archivo a enviar, una vez seleccionado el archivo damos en “abrir” y automáticamente el archivo se enviará.



El archivo se enviará al directorio que hemos especificado en el código, es importante cambiar esta dirección a una dirección existente en nuestro computador. Para cambiar este directorio debemos dirigirnos a la fuente del código:

```

Start Page  X  Servidor.java  X  Cliente.java  X
Source  Design  History
234      if (identificador == 'A') {
235          // Es un archivo
236          String nombreArchivo = recibir_mensaje.readUTF();
237
238          Date fechaActual = new Date();
239          SimpleDateFormat formatoHora = new SimpleDateFormat("HH:mm:ss");
240          String hora = formatoHora.format(fechaActual);
241          String mensajeConHora = "[" + hora + "] ";
242          txtArea.append("Cliente: " + nombreArchivo + "          "+mensajeConHora + "\n\n");
243
244          FileOutputStream fos = new FileOutputStream("C:\\Users\\HP\\Desktop\\CAR. PARA PRUEBAS\\" + nombreArchivo);
245
246          byte[] buffer = new byte[4096];
247          int bytesRead;
248          while ((bytesRead = recibir_mensaje.read(buffer)) != -1) {
249              fos.write(buffer, 0, bytesRead);
250          }
251

```