



UNIVERSIDAD  
MARIANO GÁLVEZ

# Manual de Usuario

## Proyecto Final (chat grupal)

Sistemas Operativos II  
Ingeniero: Saúl Orozco

Equipo: Proyectos #9

Pedro Daniel Vásquez Chamalé -7690-12-9149

Mario Fernando Castañeda Pérez-7690-20-145

Lester Navil Pérez Marroquin-7690-20-14011

Manuel Alejandro Sazo Linares-7690-20-13585

## Tabla de contenido

Objetivo del documento.....	3
Objetivos.....	3
Introducción .....	3
Desarrollo del Manual de Usuario .....	3
Procesos de instalación de código .....	3
Instrucciones de obtención de ip.....	8
Ejecución del programa .....	12

## Objetivo del documento

En el presente documento pretendemos mostrar al usuario el funcionamiento del proyecto el cual consiste en el uso y funcionamiento de un chat grupal.

### Objetivos

Se pretende mostrar de una manera clara y concisa el funcionamiento del chat grupal con imágenes del funcionamiento para cada proceso necesario.

## Introducción

Este proyecto consiste en ejecutar según las indicaciones solicitadas un programa que realice un chat grupal que deberá conectar 3 clientes con un solo servidor para poder chatear entre sí, el lenguaje de programación usado para este proyecto fue java.

## Desarrollo del Manual de Usuario

### Procesos de instalación de código

Lo primero que hay que instalar es el programa donde desea ejecutar el programa como por ejemplo NetBeans , IntelliJ IDEA community en nuestro caso será en IDEA.

Enlace de descarga


<https://www.jetbrains.com/idea/download/#section=windows>

← → ↻ 🔍 jetbrains.com/idea/download/#section=windows

PHP 8 y MySQL E... SQL:1.pdf Clonación RMAN si... The Network Adapt... SISTEMAS OPERATI...

**JET BRAINS** Developer Tools Team Tools Education Solutions Support Store 🔍 👤 🛒 🌐

IntelliJ IDEA What's New Features Resources Pricing **Download**



Version: 2023.1.1  
Build: 231.8770.65  
27 April 2023

## Download IntelliJ IDEA

Windows macOS Linux

**Ultimate**  
The Leading Java and Kotlin IDE

**Download** .exe ▾

Free 30-day trial available

**Community Edition**  
The IDE for pure Java and Kotlin development

**Download** .exe ▾

Free build on open source

Descargar esta versión

Ahora procedemos a instalar el JDK 20 de la siguiente página:

<https://www.oracle.com/java/technologies/javase/jdk20-archive-downloads.html>

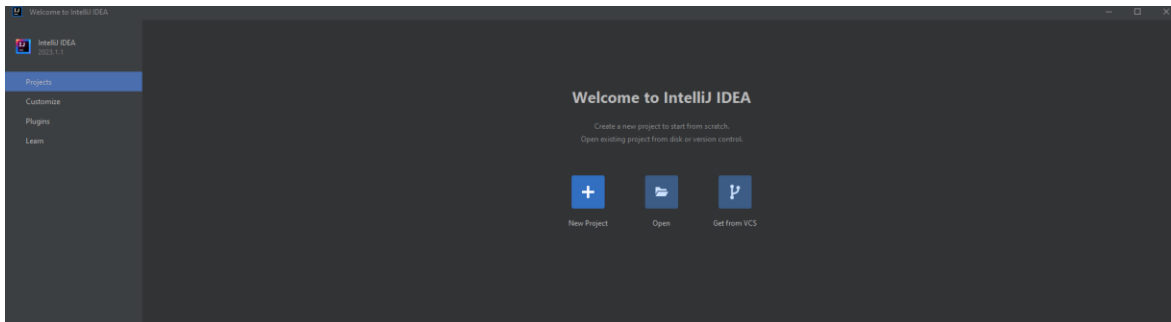
**ORACLE** Products Industries Resources Customers Partners Developers Events Company 🔍 👤 View Accounts 📄 Contact Sales

### Java SE Development Kit 20.0.1

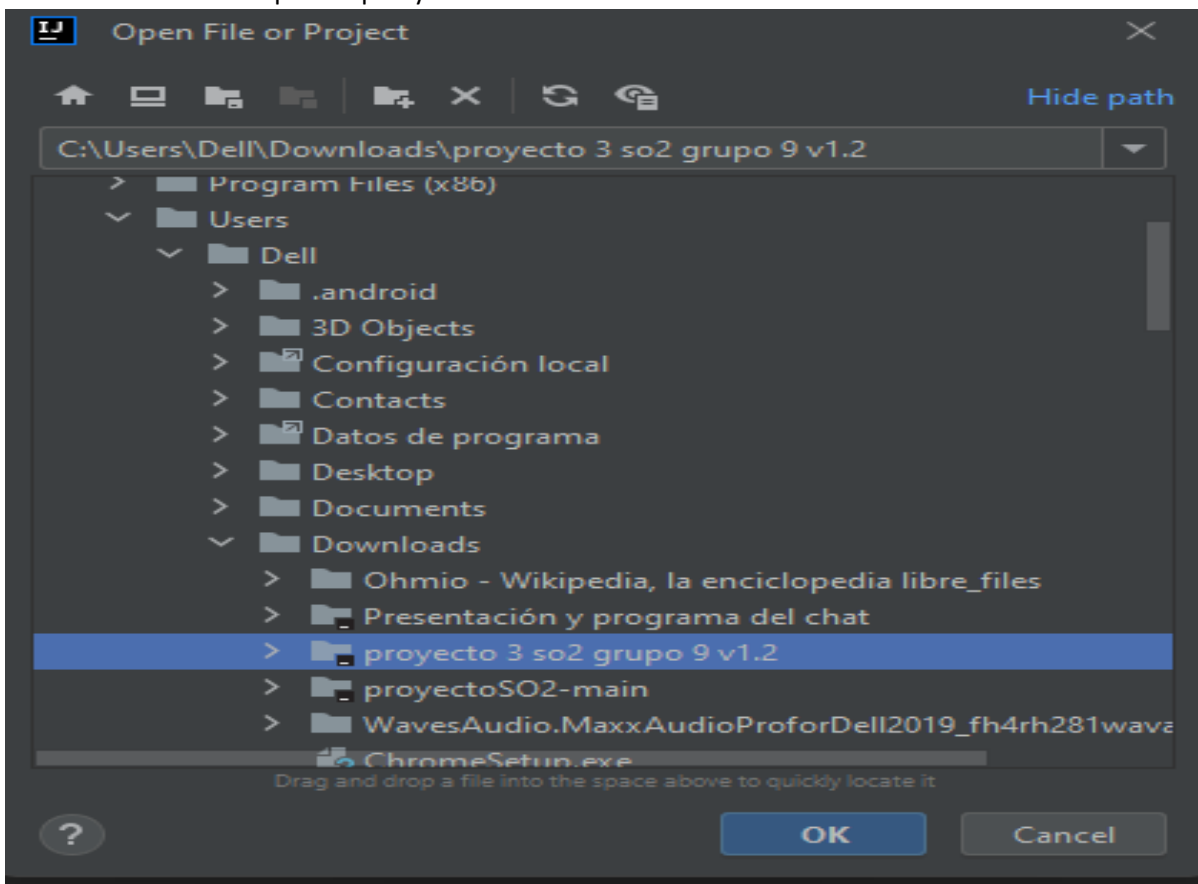
This software is licensed under the [Oracle No-Fee Terms and Conditions License](#).

Product / File Description	File Size	Download
Linux Arm 64 Compressed Archive	181.15 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-aarch64_bin.tar.gz">https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-aarch64_bin.tar.gz</a> (sha256)
Linux Arm 64 RPM Package	180.88 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-aarch64_bin.rpm">https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-aarch64_bin.rpm</a> (sha256)
Linux x64 Compressed Archive	182.69 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-x64_bin.tar.gz">https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-x64_bin.tar.gz</a> (sha256)
Linux x64 Debian Package	155.60 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-x64_bin.deb">https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-x64_bin.deb</a> (sha256)
Linux x64 RPM Package	182.41 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-x64_bin.rpm">https://download.oracle.com/java/20/archive/jdk-20.0.1_linux-x64_bin.rpm</a> (sha256)
macOS Arm 64 Compressed Archive	176.88 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-aarch64_bin.tar.gz">https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-aarch64_bin.tar.gz</a> (sha256)
macOS Arm 64 DMG Installer	176.22 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-aarch64_bin.dmg">https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-aarch64_bin.dmg</a> (sha256)
macOS x64 Compressed Archive	179.23 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-x64_bin.tar.gz">https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-x64_bin.tar.gz</a> (sha256)
macOS x64 DMG Installer	178.54 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-x64_bin.dmg">https://download.oracle.com/java/20/archive/jdk-20.0.1_macos-x64_bin.dmg</a> (sha256)
Windows x64 Compressed Archive	180.81 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_windows-x64_bin.zip">https://download.oracle.com/java/20/archive/jdk-20.0.1_windows-x64_bin.zip</a> (sha256)
Windows x64 Installer	159.95 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_windows-x64_bin.exe">https://download.oracle.com/java/20/archive/jdk-20.0.1_windows-x64_bin.exe</a> (sha256)
Windows x64 msi Installer	158.74 MB	<a href="https://download.oracle.com/java/20/archive/jdk-20.0.1_windows-x64_bin.msi">https://download.oracle.com/java/20/archive/jdk-20.0.1_windows-x64_bin.msi</a> (sha256)

Procederemos a descargar el archivo zip de nuestro programa chat grupal y procederemos a descomprimirlo en consecuencia abrimos el programa a usar en nuestro caso IntelliJ IDEA y nos aparecerá la siguiente ventana.

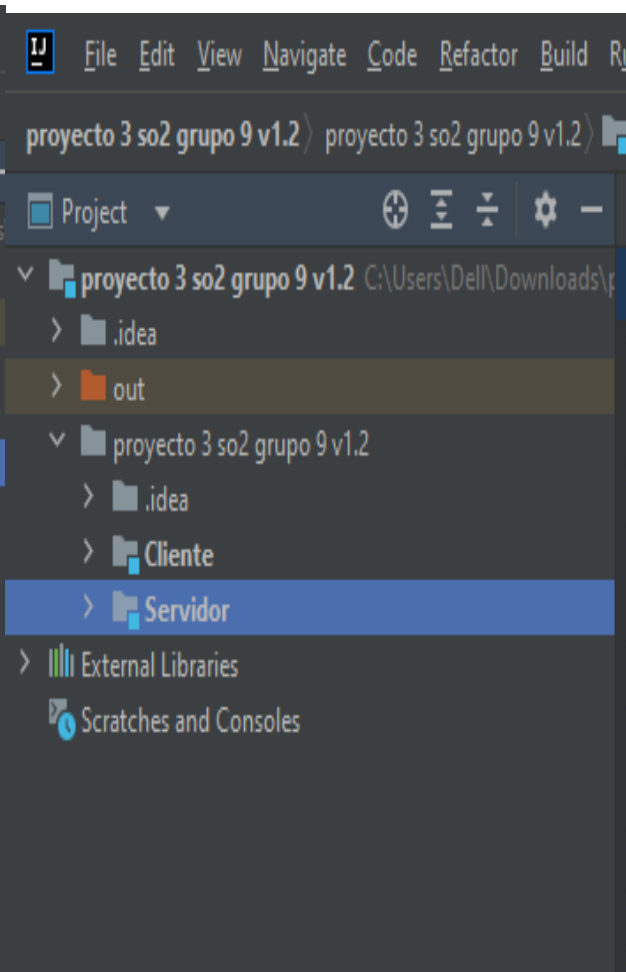
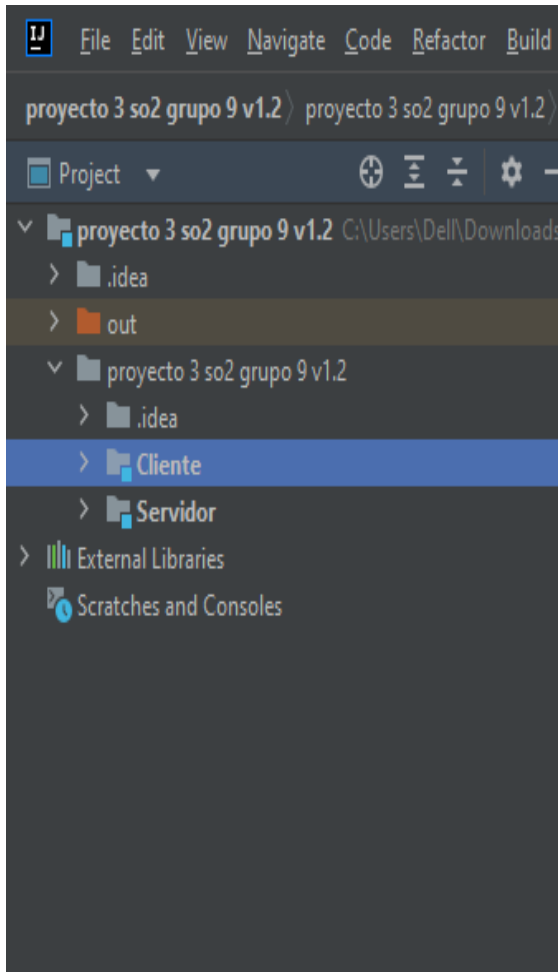


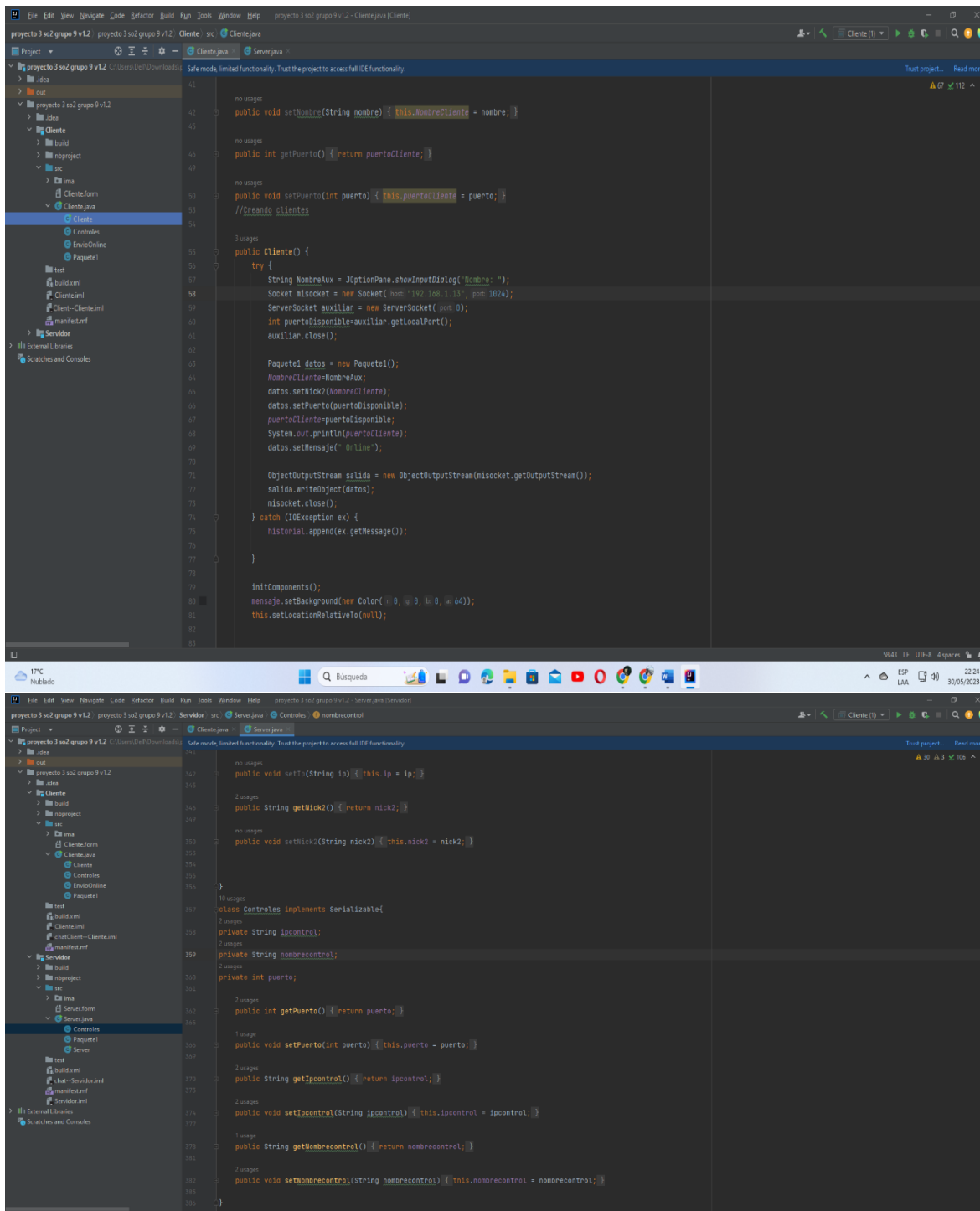
Y seleccionaremos la opción open y buscamos nuestro archivo.



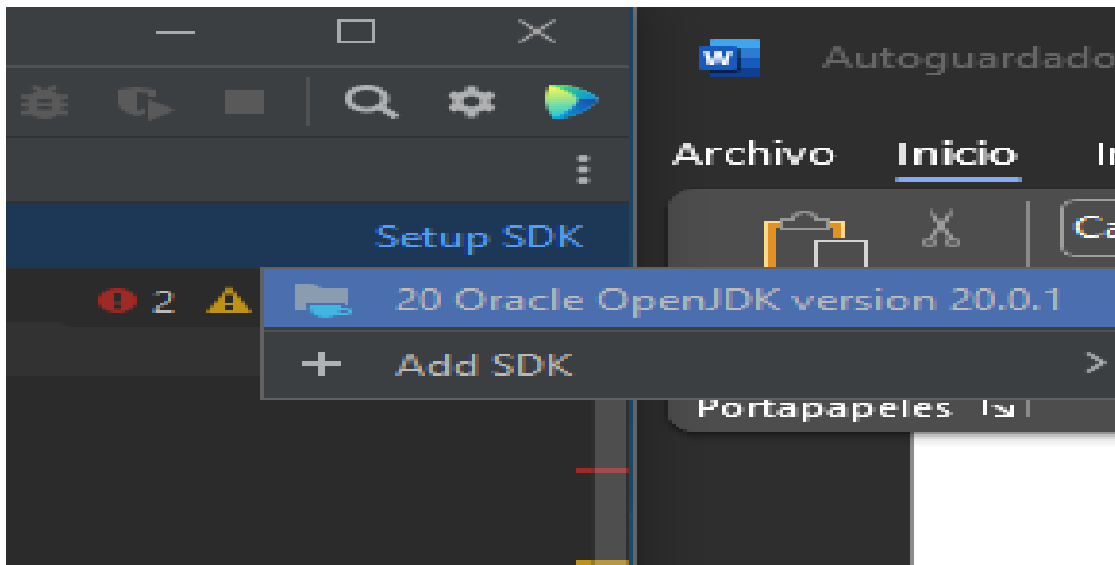
Después abrimos las carpetas de proyecto 2 so2 grupo 9v1.2

Lo siguiente será abrir las dos carpetas cliente y servidor y seleccionamos en las dos carpetas el src.



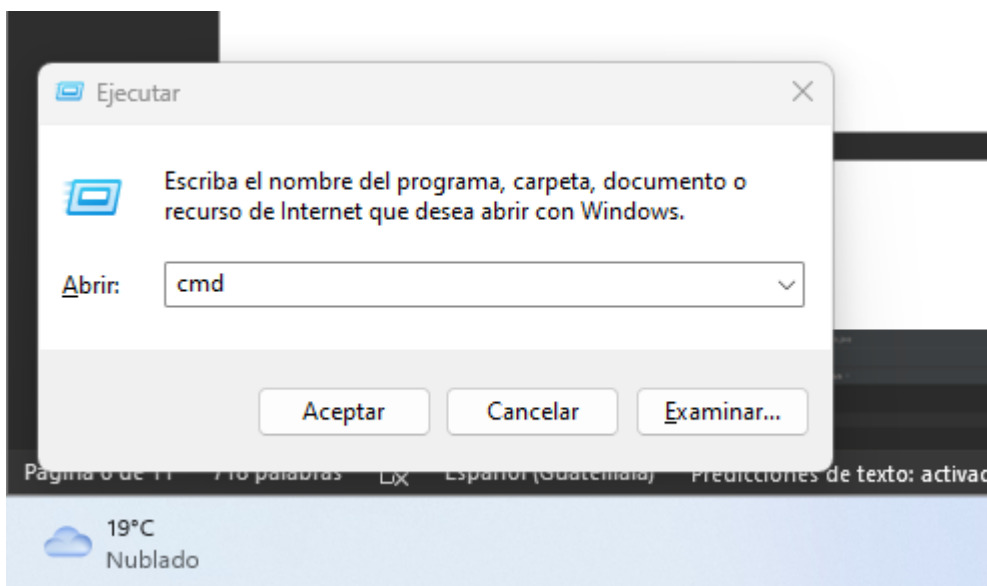


Después de abrir las 2 pestañas que nos da la cuales son cliente y servidor después de haberla realizado seleccionaremos el setup SDK que nos parecerá en el lado derecho en la parte de arriba en una esquina y seleccionaremos el 20 Oracle openJDK que previamente habíamos descargado.



### Instrucciones de obtención de ip

Después de eso nos dirigimos a la tecla Windows +R o inicio +R le damos aceptar



Ejecutamos el comando: ipconfig con esto obtendremos la dirección ip de nuestro servidor, para que nuestro cliente puede dirigirse a él para poder realizar las conexiones correspondientes.



```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Versión 10.0.22621.1702]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Dell>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . :
    Dirección IPv6 . . . . . : 2800:98:120e:c13e:ffa:7b1c:e90b:d39c
    Dirección IPv6 temporal. . . . . : 2800:98:120e:c13e:e8f4:cd9c:d3b6:a6d0
    Vínculo: dirección IPv6 local. . . . . : fe80::e644:e593:9910:ab39%14
    Dirección IPv4. . . . . : 192.168.1.13
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : fe80::1%14
                                                192.168.1.1

Adaptador de Ethernet Ethernet 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

C:\Users\Dell>
```

Con el comando en CMD estamos obteniendo la dirección de nuestro servidor, por lo que debemos copiar la Ip y reemplazar en nuestro código cliente en las siguientes secciones:

```
Run Tools Window Help proyecto 3 so2 grupo 9 v1.2 - Cliente.java [Cliente]
Cliente > src > Cliente.java > Cliente > Cliente
  Cliente.java  Server.java
Safe mode, limited functionality. Trust the project to access full IDE functionality.
35 private static int puertoCliente;
36 5 usages
37 private static String NombreCliente;
38 no usages
39 public String getNombre() { return NombreCliente; }
41 no usages
42 public void setNombre(String nombre) { this.NombreCliente = nombre; }
45 no usages
46 public int getPuerto() { return puertoCliente; }
49 no usages
50 public void setPuerto(int puerto) { this.puertoCliente = puerto; }
53 //Creando clientes
54
55 3 usages
56 public Cliente() {
57     try {
58         String NombreAux = JOptionPane.showInputDialog("Nombre: ");
59         Socket misocket = new Socket( host: "192.168.1.13", port: 1024);
60         ServerSocket auxiliar = new ServerSocket( port: 0);
61         int puertoDisponible=auxiliar.getLocalPort();
62         auxiliar.close();
63
64         Paquete1 datos = new Paquete1();
65         NombreCliente=NombreAux;
66         datos.setNick2(NombreCliente);
67         datos.setPuerto(puertoDisponible);
68         puertoCliente=puertoDisponible;
69         System.out.println(puertoCliente);
70         datos.setMensaje(" Online");
71
72         ObjectOutputStream salida = new ObjectOutputStream(misocket.getOutputStream());
73         salida.writeObject(datos);
74         misocket.close();
75     } catch (IOException ex) {
76         historial.append(ex.getMessage());
77     }
78 }
```

```
Run Tools Window Help proyecto 3 so2 grupo 9 v1.2 - Cliente.java [Cliente]
Cliente src Cliente.java Cliente enviarActionPerformed
Cliente.java Server.java
Safe mode, limited functionality. Trust the project to access full IDE functionality.
256 // TODO add your handling code here:
257 } //GEN-LAST:event_formMouseDragged
258
259 1 usage
260 private void ipdestinoActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_ipdestinoActionPerformed
261 // TODO add your handling code here:
262 } //GEN-LAST:event_ipdestinoActionPerformed
263
264 1 usage
265 private void cerrarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_cerrarActionPerformed
266 System.exit(0);
267 // TODO add your handling code here:
268 } //GEN-LAST:event_cerrarActionPerformed
269
270 1 usage
271 private void enviarActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_enviarActionPerformed
272 //Muestra ip del servidor y el puerto que van a estar abiertos
273 try {
274 Socket miSocket = new Socket( host "192.168.1.13", port 1024);
275 Paquete1 datos = new Paquete1();
276 datos.setNick2(nick.getText());
277 datos.setDestinatario(ipdestino.getSelectedItem().toString());
278 String aux;
279 aux=ipdestino.getSelectedItem().toString();
280 for(int i=0;i<listaClientes.size();i++){
281 if(aux.equalsIgnoreCase(listaClientes.get(i).getNombrecontrol())){
282 datos.setIp(listaClientes.get(i).getIpcontrol());
283 }
284 }
285 historial.append("\n-"+mensaje.getText());
286 datos.setMensaje(mensaje.getText());
287
288 ObjectOutputStream salida = new ObjectOutputStream(miSocket.getOutputStream());
289 salida.writeObject(datos);
290 mensaje.setText(null);
291 miSocket.close();
292 }
```

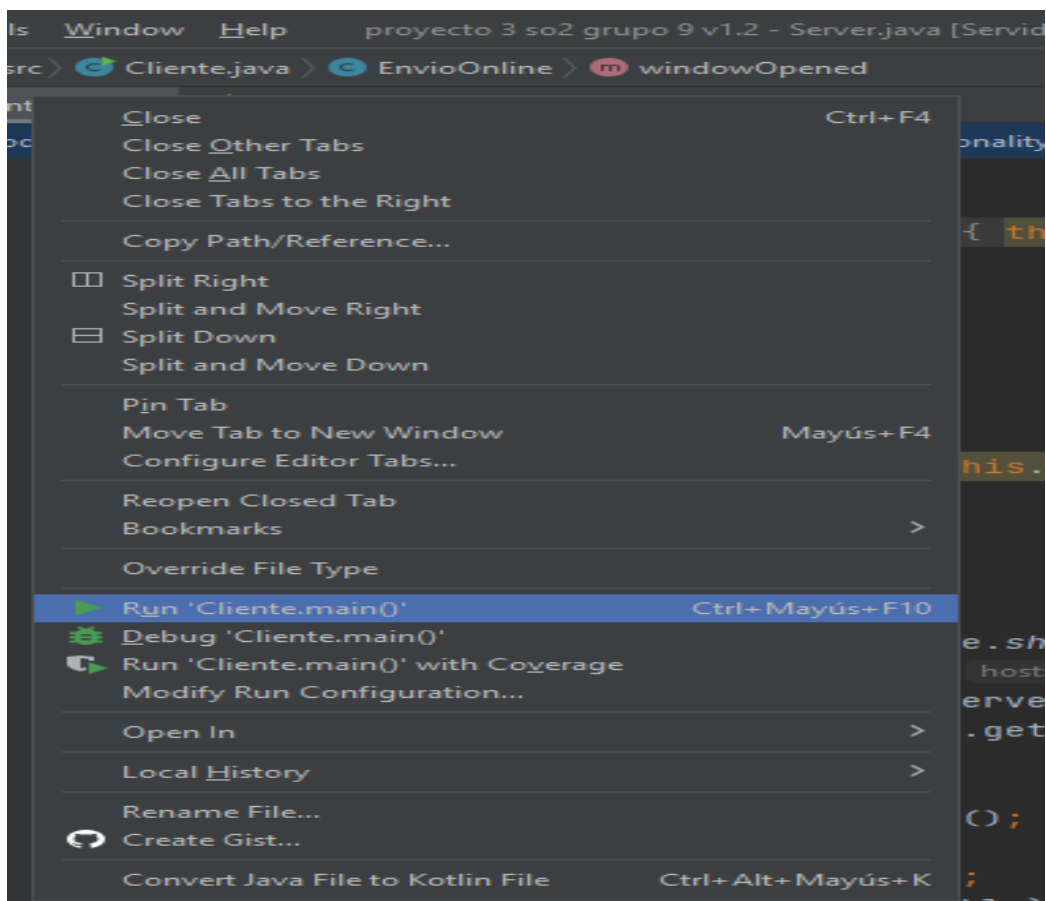
```
Run Tools Window Help proyecto 3 so2 grupo 9 v1.2 - Cliente.java [Cliente]
Cliente src Cliente.java EnvioOnline windowOpened
Cliente.java Server.java
Safe mode, limited functionality. Trust the project to access full IDE functionality.
469 3 usages
470 public void setNick2(String nick2) { this.nick2 = nick2; }
471
472
473
474
475 }
476 1 usage
477 class EnvioOnline extends WindowAdapter { //implementa todas los metodos pertenecientes a una interfaz(listener)
478 /*metodo que se ejecuta cuando se abra la ventana*/
479
480 2 usages
481 String nickx, ipserver;
482 no usages
483 public void hola (String s) { nickx=s; }
484 no usages
485 public void hola2 (String e) { ipserver=e; }
486
487 public void windowOpened(WindowEvent e){
488 try {
489 //ip del server
490
491 Socket misocket = new Socket( host "192.168.1.13", port 1024);
492
493 Paquete1 datos = new Paquete1();
494
495 datos.setNick2(nickx);
496 datos.setMensaje(" Online");
497
498 ObjectOutputStream salida = new ObjectOutputStream(misocket.getOutputStream());
499 salida.writeObject(datos);
500 misocket.close();
501
502 } catch (IOException ex) {
503 Logger.getLogger(EnvioOnline.class.getName()).log(Level.SEVERE, msg: null, ex);
504 }
505
506 }
```

Ya que configuramos la ip de nuestro servidor en nuestro programa cliente procedemos a ejecutar el servidor.

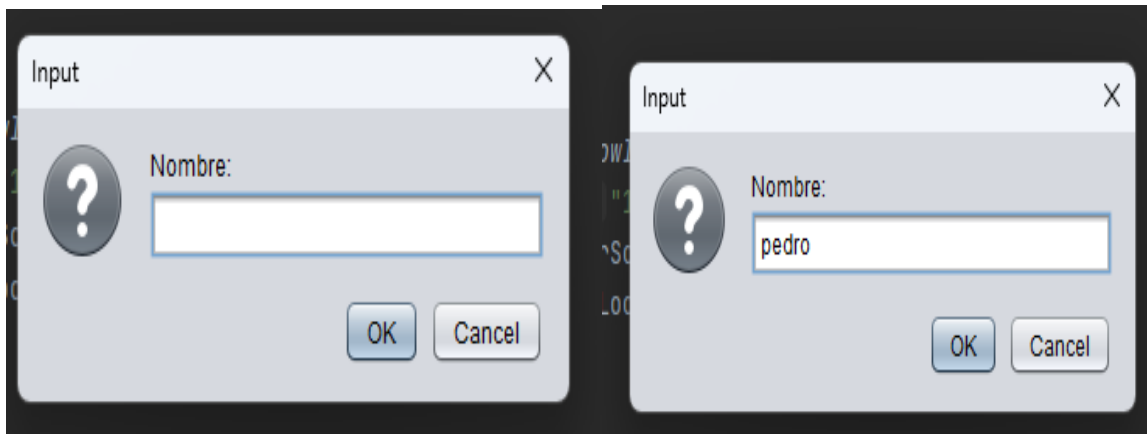
Mostrara la siguiente pestaña.



Procedemos a ejecutar cada uno de los clientes.



Nos pedirá que ingresemos nuestro nickname para poder ser identificados por el servidor y por los demás usuarios.

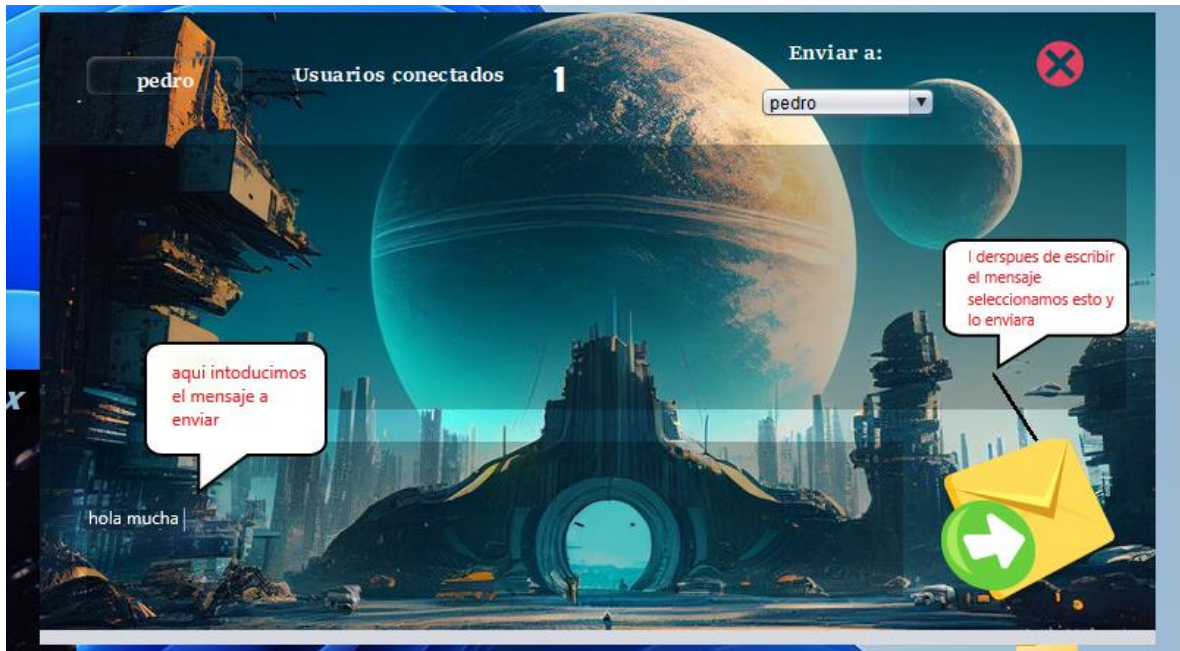


## Ejecución del programa

Después de darle a OK entraremos a nuestra interfaz la cual es de la siguiente manera:







De esta forma se mostrara a los demás usuarios conectados, algunas de sus restricciones seria la utilización de más de 80 caracteres solicitando que se despliegue en dos partes .

