



I.E.S. "ÁGORA"

Dto. Informática.

Ciclo F. de G.S.: Desarrollo de aplicaciones
multiplataforma.

PROYECTO: N.P.E.C.M.L

Autor: Alejandro Vega Serrano

Tutor: Esther Vega Rossell

Cáceres, a 15 de mayo de 2020

MANUAL DE USUARIO

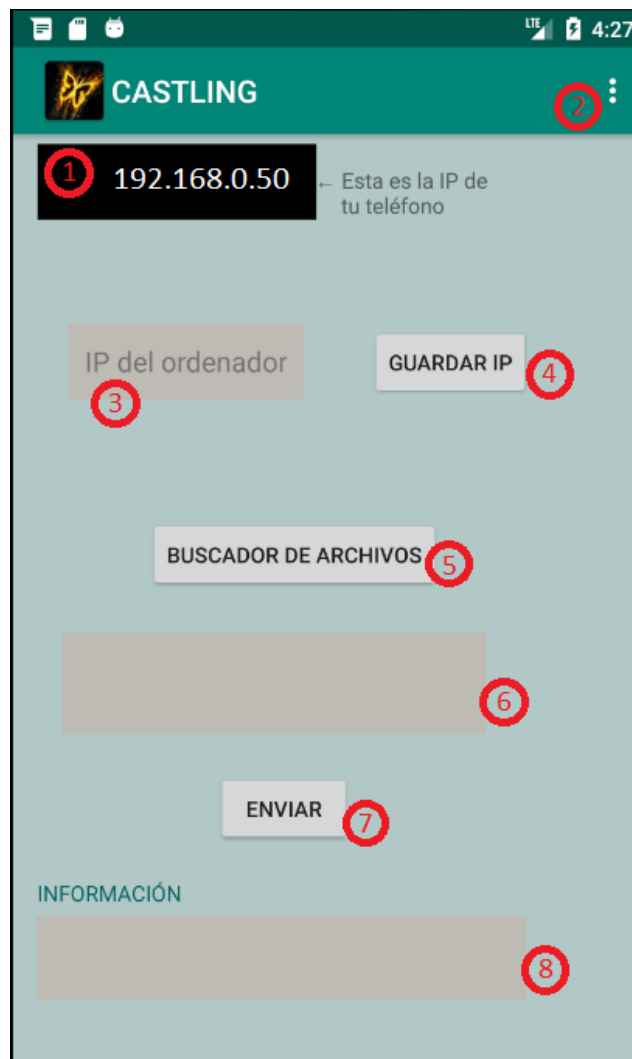
ÍNDICE

- [Introducción](#)
- [Manual de Usuario](#)
 - [Aplicación del Dispositivo Móvil](#)
 - [Aplicación del Ordenador](#)
- [Manual del Programador](#)
 - [Android](#)
 - [AndroidManifest.xml](#)
 - [Build.gradle\(Module:app\)](#)
 - [Colors.xml](#)
 - [MainActivity.java](#)
 - [ActivityMain.xml](#)
 - [Explorador de Archivos.java](#)
 - [Activity Main 2.xml](#)
 - [Instrucciones Activity.java](#)
 - [Activity Instrucciones .xml](#)
 - [Enviador Mensajes.java](#)
 - [NETBEANS](#)

Introducción

He realizado este proyecto para abordar la comunicación entre dispositivos móviles y ordenadores personales. Se me ocurrió en base para mejorar la comodidad del usuario para que haya un medio de intercambiar archivos de forma rápida entre estos dispositivos. Utilizando una comunicación por Sockets totalmente libre de cables puedes intercambiar archivos entre dispositivos conectados a una misma red. Con una interfaz sencilla, intuitiva y que ayuda al usuario regular. Quería crear una aplicación que aumentase la comodidad del usuario que utiliza en gran medida tanto el ordenador personal como el teléfono móvil.

Aplicación de Dispositivo



1. IP de tu dispositivo, debes introducirla en la aplicación de escritorio.

2. Pestaña para ver las instrucciones paso a paso de la aplicación.
3. IP del ordenador, aquí debes introducir el valor de la dirección IP que aparece en la aplicación de escritorio.
4. Botón para almacenar la IP de tu ordenador para establecerla como predeterminada para futuros usos de la aplicación.
5. Botón para seleccionar, a través de la memoria de tu teléfono, el archivo que quieres enviar a tu ordenador.
6. Archivo seleccionado con el botón “buscador de archivos”.
7. Botón que, una vez seleccionado un archivo, lo enviará a tu ordenador.
8. Aquí aparecerá información de utilidad durante el traspaso de archivos. También puedes tocar los distintos paneles para que aparezca información sobre ellos aquí.

Aplicación de Ordenador

The screenshot shows the CASTLING application interface. At the top, the word "CASTLING" is displayed in a large, bold, teal font. Below this, there are several input fields and buttons, each with a red circle and a number indicating a specific step in the process:

- 1**: Points to the input field for the computer's IP address, which contains "192.168.0.10".
- 2**: Points to the input field for the mobile device's IP address, which contains "192.168.0.50".
- 3**: Points to the button labeled "Establecer esta IP como predeterminada".
- 4**: Points to the input field for the storage path, which is currently empty.
- 5**: Points to the button labeled "Cambiar Ruta de almacenamiento".
- 6**: Points to the input field for the selected file, which is currently empty.
- 7**: Points to the button labeled "Examinar Archivo".
- 8**: Points to the button labeled "Enviar a dispositivo".

Below the input fields, there is a note: "Nota: El programa creará una subcarpeta llamada 'NPECML' en la ruta seleccionada".

A continuación, tenemos la aplicación de escritorio:

1. Dirección IP de tu ordenador que debes colocar en el campo correspondiente en tu dispositivo móvil.
2. Campo donde debes colocar la IP de tu teléfono (El cuadrado negro de la aplicación de tu dispositivo móvil).
3. Botón para establecer la IP escrita como predeterminada para futuras sesiones.
4. Ruta donde se almacenarán los archivos recibidos.
5. Botón para cambiar el lugar donde se almacenarán los archivos recibidos.
6. Archivo seleccionado.
7. Botón para seleccionar el archivo que quieres enviar a tu dispositivo.
8. Botón para, una vez seleccionado el archivo, enviarlo a tu dispositivo.

MANUAL DEL **PROGRAMADOR**

ANDROID

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.proyectoachat">

    //Permisos necesarios para el correcto funcionamiento de la
    aplicación.
    <uses-permission
        android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="CASTLING"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"

        //Campo necesario para que funcione correctamente
        android:usesCleartextTraffic="true">

        <activity android:name=".Instrucciones_Activity"></activity>
        <activity android:name=".Explorador_de_Archivos" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Build.gradle(Module:app):

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 29
    defaultConfig {
        applicationId "com.example.proyectoachat"
        minSdkVersion 23
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
        "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'

    //modulo necesario para poder enlazar el sistema Android con la
    //base de datos local
    implementation 'com.android.volley:volley:1.1.1'
}
```

colors.xml

//colores utilizados durante la creación de la aplicación.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>
    <color name="rojo">#FF0000</color>
    <color name="azul">#0000FF</color>
    <color name="celeste">#00FFFF</color>
    <color name="amarillo">#ffff33</color>
    <color name="naranja">#ff9900</color>
    <color name="violeta">#ff33ff</color>
    <color name="negro">#000000</color>
    <color name="indigo">#051094</color>
    <color name="blanco">#FFFFFF</color>
    <color name="Fondo">#B2C8C6</color>
</resources>
```



```
<color name="Paneles">#BEBAB4</color>
<color name="Letras">#006666</color>
</resources>
```

MainActivity.java

```
package com.example.proyectoachat;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.provider.MediaStore;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.ObjectInputStream;
import java.net.Inet4Address;
import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.ServerSocket;
```

```

import java.net.Socket;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Collections;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;

public class MainActivity extends AppCompatActivity {

    //puerto predeterminado que usará Android para recibir los
    archivos.
    final int Puerto_Entrada=6001;

    //variable usada para la confirmación de permisos.
    private static final int REQUEST_CODE_ASK_PERMISSION = 111;

    //conexiones con el layout principal de la aplicación.
    ConstraintLayout FONDO;
    static TextView Archivo_Sel,textoip,INFO;
    static EditText IP_Ordenador;
    Button EditarIP,btn_Enviar,btn_Buscador_Archivos;
    TextView auxiliar;

    static boolean Operacion_Correcta=false;
    static String Nombre_archivo_Enviar=null,
    Ruta_Archivo=null,IP_Computadora;

    RequestQueue requestQueue;

    /*Método onCreate que inicializa la aplicación. El método realiza
    las siguientes acciones:
    -Cambia el icono de la aplicación.
    -Solicita los permisos necesarios para ejecutar la aplicación.
    -Asocia los controles del layout a los de la aplicación.
    -Asigna métodos "onClick" a los campos de texto del layout para
    que aparezca un texto en el campo de INFO para darle más información
    al usuario.
    -Llama a los archivos php alojados en el ordenador para ejecutar
    sentencias SQL y poder interactuar con la base de datos local que
    almacena la información de la aplicación, así como la IP
    predeterminada, la ruta de almacenamiento de los archivos o un
    registro de los archivos enviados y recibidos de forma exitosa.
    -Iniciar el hilo que permitirá la recepción de los archivos.
    */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Poner el icono en el Action Bar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setIcon(R.mipmap.ic_launcher);

        solicitarPermisos();
    }

```

```

FONDO=findViewById(R.id.FONDO);
auxiliar=findViewById(R.id.textView);
btn_Buscador_Archivos=findViewById(R.id.btn_Buscador);
btn_Enviar=findViewById(R.id.btn_Envio);
Archivo_Sel=findViewById(R.id.AM1_Archivo_seleccionado);
IP_Ordenador=findViewById(R.id.txt_IP_Computador);
textoip=findViewById(R.id.AM1_IP);

EditarIP=findViewById(R.id.btn_guardar_IP);

textoip.setText(getIP());

textoip.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        INFO.setText("Esta es la IP de tu dispositivo, debes
colocarla en la aplicacion de tu ordenador");
    }
});

Archivo_Sel.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(Archivo_Sel.getText().toString().length()==0){
            INFO.setText("Aquí se mostrará el archivo a
enviar. Pulsa 'Buscador de Archivos' para seleccionar un archivo para
transferirlo.");
        }else{
            INFO.setText("Esta es la ruta del archivo a
enviar. Pulsa 'Enviar' para iniciar la transferencia.");
        }
    }
});

INFO=findViewById(R.id.txt_INFO);

recuperarINFO("http://192.168.1.192:8080/NPECML/recuperar.php");

EditarIP.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        IP_Computadora=IP_Ordenador.getText().toString();
Guardar_IP("http://"+IP_Computadora+":8080/NPECML/editar.php");
    }
});

Thread myThread= new Thread(new MyServerThread());
myThread.start();

}

/*
Método para crear el menú de la esquina superior derecha de la
aplicación que mostrará el menú de "Instrucciones" para que el usuario

```

pueda acceder de forma fácil y discreta a la pantalla de las instrucciones de la aplicación.

El método solamente creará un único menú que será este "Instrucciones" y con una llamada a la activity que gestiona la pantalla de las Instrucciones.

```
*/
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.MnuOpcl:
            Intent intent=new
Intent(MainActivity.this, Instrucciones_Activity.class);
            startActivity(intent);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

/*      Método para solicitar los permisos de la aplicación
(previamente escritos en el Manifest) al usuario en tiempo real de
ejecución.
*/
private void solicitarPermisos() {
    int permisoWriteExternalStorage, permisoReadExternalStorage,
permisoInternet;
    permisoReadExternalStorage=
ActivityCompat.checkSelfPermission(MainActivity.this,Manifest.permission.
READ_EXTERNAL_STORAGE);
    permisoInternet
=ActivityCompat.checkSelfPermission(MainActivity.this,Manifest.permission.
INTERNET);

permisoWriteExternalStorage=ActivityCompat.checkSelfPermission(MainAct
ivity.this,Manifest.permission.WRITE_EXTERNAL_STORAGE);
    if (permisoInternet!=PackageManager.PERMISSION_GRANTED ||
permisoReadExternalStorage!=PackageManager.PERMISSION_GRANTED ||
permisoWriteExternalStorage!=PackageManager.PERMISSION_GRANTED) {
        if (Build.VERSION.SDK_INT>=Build.VERSION_CODES.M) {
            requestPermissions(new
String[]{Manifest.permission.INTERNET,Manifest.permission.WRITE_EXTERN
AL_STORAGE,Manifest.permission.READ_EXTERNAL_STORAGE},REQUEST_CODE_ASK
_PERMISSION);
        }
    }
}

/*      Método que llama a la base de datos local para recuperar la
información necesaria para el comienzo de la aplicación con un
sencillo proceso de JSONArrayRequest para almacenar la información
recibida con las sentencias SQL que realiza el archivo php. En este
```

caso recupera la dirección IP que tengas almacenada como predeterminada para que el usuario no tenga que introducirla cada vez que inicia la app

```
*/
    public void recuperarINFO(String URL){
        JsonRequest jsonArrayRequest=new JsonRequest(URL,
new Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {
                JSONObject jsonObject = null;
                for (int i = 0; i < response.length(); i++) {
                    try {
                        jsonObject = response.getJSONObject(i);

IP_Ordenador.setText(jsonObject.getString("IP_Predeterminada"));
                    } catch (JSONException e) {
                        Toast.makeText(getApplicationContext(),
e.getMessage(), Toast.LENGTH_LONG).show();
                    }
                }
            }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(getApplicationContext(), "Error de
conexion", Toast.LENGTH_LONG).show();
                }
            });
        requestQueue= Volley.newRequestQueue(this);
        requestQueue.add(jsonArrayRequest);
    }

    //Método que usará el cliente para abrir el buscador de archivos
    public void Buscador_Archivos(View v){
        Intent intent=new Intent(this, Explorador_de_Archivos.class);
        startActivity(intent);
    }

    /* Método que, una vez seleccionado el archivo a enviar, recupera la
    información necesaria de este archivo (nombre y la ruta donde está
    almacenado) para, posteriormente, enviar esta información a la clase
    "Enviador_Mensajes" que procederá a enviar el archivo
    */
    public void Enviar(View v){
        IP_Computadora=IP_Ordenador.getText().toString();
        if(IP_Computadora.length()==0){
            Toast.makeText(this, "Debes introducir la IP de tu
ordenador (Se muestra en la aplicacion de
escritorio)", Toast.LENGTH_SHORT).show();
        }else{
            if(Ruta_Archivo==null || Nombre_archivo_Enviar==null){
                INFO.setText("Selecciona un archivo a enviar con el
boton\n 'Buscador de archivos'");
            }else{
                Enviador_Mensajes enviador_mensajes=new
Enviador_Mensajes();

                enviador_mensajes.execute(Ruta_Archivo, Nombre_archivo_Enviar);
            }
        }
    }
}
```

```

        if(Operacion_Correcta) {

ejecutarServicio("http://" + IP_Computadora + ":8080/NPECML/insertar_trans
accion.php");
        }
        Operacion_Correcta=false;
        Nombre_archivo_Enviar=null;
        Ruta_Archivo=null;
    }

}

}

/*      Método para obtener la IP de tu dispositivo automáticamente y
que aparezca en el campo de la IP del dispositivo del layout
principal. Utiliza NetworkInterface para obtener esta dirección IP.
*/
    public static String getIP() {
        List<InetAddress> addrs;
        String address = "";
        try{
            List<NetworkInterface> interfaces =
Collections.list(NetworkInterface.getNetworkInterfaces());
            for(NetworkInterface intf : interfaces){
                addrs = Collections.list(intf.getInetAddresses());
                for(InetAddress addr : addrs){
                    if(!addr.isLoopbackAddress() && addr instanceof
Inet4Address){
                        address =
addr.getHostAddress().toUpperCase(new Locale("es", "ES"));
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        textoip.setText(address);
        return address;
    }

    public boolean isExternalStorageWritable() {
        String state = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(state)) {
            return true;
        }
        return false;
    }

/*      Método que almacena el texto escrito en el campo de "IP del
ordenador" en la base de datos para que, posteriormente, la aplicación
coja este valor automáticamente y así mejorar la usabilidad y
comodidad de la aplicación. El método llama a un archivo php que
ejecuta la sentencia SQL necesaria para almacenar estos valores.
*/
    public void Guardar_IP(String url){

```

```

        if(IP_Ordenador.getText().toString().trim().length()!=0){
            final String
Nueva_IP_Predeterminada=IP_Ordenador.getText().toString();
            StringRequest stringRequest=new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
                @Override
                public void onResponse(String response) {

                }
            }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {

                }
            }) {
                @Override
                protected Map<String, String> getParams() throws
AuthFailureError {
                    Map<String,String> parametros=new
HashMap<String,String>();

                    parametros.put("IP_Predeterminada",Nueva_IP_Predeterminada);
                    return parametros;
                }
            };
            requestQueue= Volley.newRequestQueue(this);
            requestQueue.add(stringRequest);
            Toast.makeText(this,"Se ha guardado la nueva IP como
predeterminada",Toast.LENGTH_SHORT).show();

        }else{
            Toast.makeText(this,"Debes introducir la IP de tu
ordenador primero",Toast.LENGTH_LONG).show();
        }

    }

    /*      Clase que se usará para la obtención de fecha y hora del
dispositivo
    */
    static class Utiles {

        /*      Métodos que recuperan la información de la fecha y hora del
dispositivo para, posteriormente, almacenarlo en un campo de "fecha"
que tiene la base de datos y así incluirlo en el registro de archivos
intercambiados.
        */

        public static String obtenerHoraActual(String zonaHoraria) {
            String formato = "HH:mm:ss";
            return Utiles.obtenerFechaConFormato(formato,
zonaHoraria);
        }

        public static String obtenerFechaActual(String zonaHoraria) {
            String formato = "yyyy-MM-dd";
            return Utiles.obtenerFechaConFormato(formato,
zonaHoraria);
        }
    }

```

```

    }

    /* Método para obtener la fecha y la hora en base a los formatos
    elegidos
    */
    @SuppressWarnings("SimpleDateFormat")
    public static String obtenerFechaConFormato(String formato,
String zonaHoraria) {
        Calendar calendar = Calendar.getInstance();
        Date date = calendar.getTime();
        SimpleDateFormat sdf;
        sdf = new SimpleDateFormat(formato);
        sdf.setTimeZone(TimeZone.getTimeZone(zonaHoraria));
        return sdf.format(date);
    }

    /*      Método que almacena la fecha y hora en la base de datos local
    obtenidos con los métodos anteriores y así completar el registro.
    */
    public void ejecutarServicio(String url){

        final String fecha=Utiles.obtenerFechaActual("GMT+2")+ " "+
Utiles.obtenerHoraActual("GMT+2");

        StringRequest stringRequest=new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        }) {
            @Override
            protected Map<String, String> getParams() throws
AuthFailureError {
                Map<String,String> parametros=new
HashMap<String,String>();
                parametros.put("nombre",Nombre_archivo_Enviar);
                parametros.put("fecha",fecha);
                parametros.put("origen","Dispositivo");
                return parametros;
            }
        };
        requestQueue= Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }
}

```



```

    /*Clase que implementa el servidor que recibirá archivos del
ordenador
*/
    class MyServerThread implements Runnable{

        Socket s_recibir1;
        Socket s_recibir2;

        ServerSocket ss;

        ObjectInputStream ois;
        FileOutputStream fichero_nuevo;

        InputStreamReader isr;
        BufferedReader bf;

        String Nombre_Archivo_Recibir="dafault";

        /*Método que mediante la conexión por sockets, cogerá la
información transferida desde el ordenador a nuestro dispositivo móvil
y almacenará esta información en un archivo idéntico que el que había
originalmente.
*/
        @Override
        public void run() {

            //Si están los permisos de la aplicación dados podemos
comenzar la creación del servidor.
            if(isExternalStorageWritable()){
                try {
                    ss=new ServerSocket(Puerto_Entrada);//Usamos el puerto
de entrada para crear el server.

                    while (true){
                        s_recibir1=ss.accept();

                        /*Accedemos a la información que hemos recibido
por el InputStream del Socket, en este primer recibo obtenemos el
nombre y la extensión del archivo enviado.
*/
                        isr=new
InputStreamReader(s_recibir1.getInputStream());
                        bf=new BufferedReader(isr);
                        Nombre_Archivo_Recibir=bf.readLine();
                        s_recibir1.close();

                        /*Accedemos nuevamente al InputStream del Socket
para recibir una segunda tanda de datos los cuales contendrán el
archivo en un Array de bytes.
*/
                        s_recibir2=ss.accept();
                        ois = new
ObjectInputStream(s_recibir2.getInputStream());
                        Object Almacenamiento = ois.readObject();
                        int auxiliar[] = (int[]) Almacenamiento;

```

```

        /*Una vez tenemos este Array de bytes parseados y
        almacenados procedemos a la creación (si no existe anteriormente) de
        una subcarpeta que contendrá todos los archivos recibidos.
        */
        File TarjetaSD= new
File(Environment.getExternalStorageDirectory().getAbsolutePath());

        File CarpetaProyecto=new
File(TarjetaSD.getPath()+"/NPECML/");

        if (!CarpetaProyecto.exists()){
            CarpetaProyecto.mkdirs();
        }

        /*Una vez sepamos que la carpeta del
        almacenamiento está completa procedemos a crear un archivo nuevo, cuyo
        nombre será el nombre que hemos recibido en la primera tanda de datos
        y su contenido será los datos obtenidos en el Array de bytes que hemos
        obtenido en la segunda tanda
        */
        File aux=new
File(CarpetaProyecto.getPath(),Nombre_Archivo_Recibir);
        aux.createNewFile();

        fichero_nuevo = new FileOutputStream(aux);

        INFO.setText("Archivo "+Nombre_Archivo_Recibir+"
recibido, procediendo al almacenamiento...");

        for (int i = 0; i < auxiliar.length; i++) {
            fichero_nuevo.write(auxiliar[i]);
        }

        INFO.setText("Archivo almacenado con éxito");

        fichero_nuevo.close();

        //-----
        -----

        ois.close();
        s_recibir2.close();

    }

} catch (IOException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}

}

}}}

```

Activity main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FONDO"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/Fondo"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btn_Envio"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="132dp"
        android:layout_marginBottom="144dp"
        android:onClick="Enviar"
        android:text="Enviar"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/btn_Buscador"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="88dp"
        android:layout_marginBottom="308dp"
        android:onClick="Buscador_Archivos"
        android:text="Buscador de Archivos"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/btn_guardar_IP"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="232dp"
        android:layout_marginBottom="432dp"
        android:text="Guardar IP"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/txt_IP_Computador"
        android:layout_width="153dp"
        android:layout_height="49dp"
        android:layout_marginStart="36dp"
        android:layout_marginBottom="432dp"
```

```
    android:ems="10"
    android:hint="IP del ordenador"
    android:inputType="textPersonName"
    android:textColor="@color/blanco"
    android:background="@color/Paneles"
    android:textAlignment="center"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
    android:id="@+id/AM1_IP"
    android:layout_width="181dp"
    android:layout_height="49dp"
    android:layout_marginStart="16dp"
    android:layout_marginBottom="548dp"
    android:background="@color/negro"
    android:gravity="center"
    android:textAlignment="center"
    android:textColor="@color/blanco"
    android:textSize="24dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
    android:id="@+id/AM1_Archivo_seleccionado"
    android:layout_width="275dp"
    android:layout_height="66dp"
    android:layout_marginStart="32dp"
    android:layout_marginBottom="216dp"
    android:background="@color/Paneles"
    android:gravity="center"
    android:textAlignment="center"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
    android:id="@+id/txt_INFO"
    android:layout_width="317dp"
    android:layout_height="55dp"
    android:layout_marginStart="16dp"
    android:layout_marginBottom="44dp"
    android:gravity="center"
    android:background="@color/Paneles"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginBottom="104dp"
    android:text="INFORMACIÓN"
    android:textColor="@color/Letras"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```

        android:layout_marginStart="196dp"
        android:layout_marginBottom="548dp"
        android:text="← Esta es la IP de\n      tu teléfono"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Explorador de Archivos.java

```

package com.example.proyectochat;

import android.app.Activity;
import android.app.ListActivity;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Explorador_de_Archivos extends ListActivity {
    private List listaNombresArchivos;
    private List listaRutasArchivos;
    private ArrayAdapter adaptador;
    private String directorioRaiz;
    private TextView carpetaActual;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_2);

        carpetaActual=(TextView) findViewById(R.id.Actual);
        directorioRaiz=
Environment.getExternalStorageDirectory().getPath();

        verArchivosDirectorio(directorioRaiz);

    }

    private void verArchivosDirectorio(String rutaDirectorio) {
        carpetaActual.setText("Estas en: " + rutaDirectorio);
        listaNombresArchivos = new ArrayList();
        listaRutasArchivos = new ArrayList();
    }
}

```

```

File directorioActual = new File(rutaDirectorio);
File[] listaArchivos = directorioActual.listFiles();

int x = 0;

// Si no es nuestro directorio raiz creamos un elemento que
nos
// permita volver al directorio padre del directorio actual
if (!rutaDirectorio.equals(directorioRaiz)) {
    listaNombresArchivos.add("../");
    listaRutasArchivos.add(directorioActual.getParent());
    x = 1;
}

// Almacenamos las rutas de todos los archivos y carpetas del
directorio
for (File archivo : listaArchivos) {
    listaRutasArchivos.add(archivo.getPath());
}

// Ordenamos la lista de archivos para que se muestren en
orden alfabetico
Collections.sort(listaRutasArchivos,
String.CASE_INSENSITIVE_ORDER);

// Recorremos la lista de archivos ordenada para crear la
lista de los nombres
// de los archivos que mostraremos en el listView
for (int i = x; i < listaRutasArchivos.size(); i++){
    File archivo = new File((String)
listaRutasArchivos.get(i));
    if (archivo.isFile()) {
        listaNombresArchivos.add(archivo.getName());
    } else {
        listaNombresArchivos.add("/") + archivo.getName());
    }
}

// Si no hay ningún archivo en el directorio lo indicamos
if (listaArchivos.length < 1) {
    listaNombresArchivos.add("No hay ningun archivo");
    listaRutasArchivos.add(rutaDirectorio);
}

// Creamos el adaptador y le asignamos la lista de los nombres
de los
// archivos y el layout para los elementos de la lista
adaptador = new ArrayAdapter(this,
R.layout.text_view_lista_archivos, listaNombresArchivos);
setListAdapter(adaptador);
}

@Override
protected void onItemClick(ListView l, View v, int position,
long id) {

    File archivo=new File ((String)
listaRutasArchivos.get(position));

```

```

        // Si es un archivo se muestra un Toast con su nombre y si es
        un directorio
        // se cargan los archivos que contiene en el listView
        if (archivo.isFile()) {

            MainActivity.Nombre_archivo_Enviar=archivo.getName();
            MainActivity.Ruta_Archivo=archivo.getAbsolutePath();

            MainActivity.Archivo_Sel.setText("Se ha seleccionado:\n "+
            archivo.getName());

            finish();

        } else {
            // Si no es un directorio mostramos todos los archivos que
            contiene

            verArchivosDirectorio(String.valueOf(listaRutasArchivos.get(position))
            );

        }

    }

}

```

Activity Main 2.xml

/*Esta activity contendrá el layout que usará la pestaña de "buscador de archivos, compuesto por un campo de texto que contendrá la ruta actual de donde se encuentra el usuario y un listview con la lista de carpetas y archivos que contenga el directorio actual.

```

*/
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/Fondo"
tools:context=".MainActivity">

    <TextView
        android:id="@+id/Actual"
        android:layout_width="396dp"
        android:layout_height="74dp"
        android:text="J"
        android:textAlignment="center"
        android:textSize="20dp"
        android:background="@color/Paneles"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.023" />

```

```

<ListView
    android:id="@android:id/list"
    android:layout_width="406dp"
    android:layout_height="580dp"
    android:background="@color/Paneles"
    android:divider="@color/indigo"
    android:dividerHeight="2dp"
    android:foregroundGravity="bottom"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"></ListView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Instrucciones Activity.java

```

/*
package com.example.proyectochat;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class Instrucciones_Activity extends AppCompatActivity {

    final String Instrucciones="INSTRUCCIONES:\n"
        +"1. Comprueba que la aplicación de CASTLING de tu
ordenador esté abierta.\n"
        +"2. Introduce la IP de tu ordenador (la cual aparece en
la aplicación de tu ordenador) en el campo de texto 'IP del
ordenador'.\n"
        +"3. Asegúrate de que la IP de tu dispositivo (la cual
aparece en el cuadrado negro) esté correctamente escrita en la
aplicación de CASTLING de tu ordenador.\n"
        +"4. Selecciona el archivo que quieres transferir con el
botón 'Buscador de Archivos'. \n"
        +"5. Presiona el botón 'Enviar' para enviar el archivo
seleccionado a tu ordenador.";
    TextView texto_instrucciones;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_instrucciones);
        texto_instrucciones=findViewById(R.id.txt_instrucciones);
        texto_instrucciones.setText(Instrucciones);
    }
}

```

activity instrucciones .xml

```

//Este activity contendrá el layout para la pantalla de instrucciones.

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout

```



```

xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Instrucciones_Activity">

    <TextView
        android:id="@+id/txt_instrucciones"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="0dp"
        android:background="@color/negro"
        android:textColor="@color/blanco"
        android:textSize="23dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Enviador Mensajes.java

```

package com.example.proyectochat;

import android.annotation.SuppressLint;
import android.content.Context;
import android.os.AsyncTask;
import android.widget.Toast;

import java.io.DataOutputStream;
import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.PrintWriter;
import java.net.Socket;

public class Enviador_Mensajes extends AsyncTask<String,String,Void> {

    Socket s1;
    Socket s2;

    ObjectOutputStream tuberia_oos;

```

```

PrintWriter pw;

//Puerto predeterminado que usará la app para enviar archivos.
final int Puerto_Salida=6000;

/*
    Método que una vez obtenido el archivo que queremos enviar
    mediante el "buscador de archivos", pasamos por parámetros el nombre y
    la ruta de dicho archivo para ejecutar el envío a nuestro ordenador.
*/
    @SuppressWarnings("WrongThread")
    @Override
    protected void doInBackground(String... voids) {

        File origen=new File(voids[0]); //Ruta del archivo pasada por
        parámetro.

        String Nombre_Archivo=voids[1]; //Nombre del archivo pasado por
        parámetro.

        //Contador para las posiciones del Array
        int contador=0;

        //Array de enteros que contendrá la información en bytes de
        nuestro archivo a enviar.
        int datos_entrada[];

        if(origen.exists()) {
            try {
                //Seleccionar archivo

                FileInputStream in =new FileInputStream(origen);
                long tamalo=in.getChannel().size();
                datos_entrada=new int[(int)tamalo];

                s1 = new Socket(MainActivity.IP_Computadora,
                Puerto_Salida);

                MainActivity.INFO.setText("Enviando archivo: " +
                Nombre_Archivo + " porfavor, espere...");

                /*En primer lugar mandamos una señal de información
                con el nombre del archivo y su extensión
                */

                pw = new PrintWriter(s1.getOutputStream());
                pw.write(Nombre_Archivo);
                pw.flush();
                pw.close();
                s1.close();

                /*Una vez el proceso anterior se ha completado
                correctamente significará que la conexión entre el dispositivo móvil y
                el ordenador está operativa, por lo que podemos proceder a separar el
                archivo seleccionado en bytes y almacenarlos en "datos_entrada".
                */

                while(true) {

                    int byte entrada = in.read();

```

```

        if(byte_entrada!=-1){
            datos_entrada[contador]=byte_entrada;
        }else{
            break;
        }

        contador++;
    }

    s2 = new Socket(MainActivity.IP_Computadora,
    Puerto_Salida);

    /*Una vez tenemos el array de datos procedemos a
    parsear la información como un Object, obtenemos el OutputStream del
    Socket y lo asociamos a un ObjectOutputStream y ya solo quedaría
    enviar la información almacenada en el Object a través del
    ObjectOutputStream.
    */

    tuberia_oos = new
    ObjectOutputStream(s2.getOutputStream());

    Object Almacenamiento = datos_entrada;

    tuberia_oos.writeObject(Almacenamiento);

    tuberia_oos.flush();

    /*Por último cerramos todas las conexiones que
    queden abiertas.*/
    tuberia_oos.close();

    s2.close();

    MainActivity.INFO.setText("Archivo enviado con
    éxito");

    MainActivity Operacion_Correcta=true;

    in.close();
    if(!s1.isClosed()){
        s1.close();
    }
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (EOFException E) {
    E.printStackTrace();
} catch (IOException e){
    MainActivity.INFO.setText("Error en la conexion,
    comprueba que la IP del\n ordenador esté bien escrita y que la
    aplicacion del ordenador esté abierta");
}

}

return null;
}

}

```

NETBEANS

ProyectoServer.java

```
import com.mysql.jdbc.Statement;
import java.awt.Image;
import java.awt.Toolkit;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.ObjectInputStream;
import java.io.ObjectOutput;
import java.io.ObjectOutputStream;
import java.io.PrintWriter;
import java.net.ConnectException;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Calendar;
```

```
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
```

```
/**
 *
```

```

    * @author Usuario
    */
public class ProyectoServer extends javax.swing.JFrame {

    /**
     * Creates new form MyServerFrame
     */

    boolean archivo_encontrado=false;
    String ruta;

    static String Ruta_Almacenamiento="";

    //-----//

    static String IP_DISPOSITIVO=null;
    static String IP_SERVIDOR=null;

    //Puertos predeterminados para la enviar y recibir archivos.
    static final int PUERTO_RECIBIR=6000;
    static final int PUERTO_ENVIAR=6001;

    Connection conexion;
    Statement sentencia;

    // Aqui obtenemos la ip local de la maquina
    public static void ObtenerIP() throws Exception {
        InetAddress address = InetAddress.getLocalHost();

        IP_SERVIDOR=address.getHostAddress();

    }
    private String Nombre_Archivo_Enviar="default";

    public ProyectoServer() {
        initComponents();
        this.setTitle("PROYECTO: NPECML");
        PrepararBaseDatos();
        Cargar_Configuracion();
    }

    @Override
    public Image getIconImage() {
        Image retValue =
Toolkit.getDefaultToolkit().getImage(ClassLoader.getSystemResource("resources/iconoapp.png"));
        return retValue;
    }

    private void PrepararBaseDatos(){
        //1.- Cargar el controlador
        try
        {
            String controlador="com.mysql.jdbc.Driver";

```

```

        Class.forName(controlador).newInstance();

    } catch (Exception ex)
    {
        JOptionPane.showMessageDialog(null, "Error al cargar
el controlador.");
    }

//2.- Crear el objeto conexión.

    String DBURL="jdbc:mysql://localhost/p_npecml";
    String usuario="root";
    String password="";
    try
    {

conexion=DriverManager.getConnection(DBURL,usuario,password);
    } catch (SQLException ex)
    {

Logger.getLogger(ProyectoServer.class.getName()).log(Level.SEVERE,
null, ex);
    }
    try
    {

//3.- Crear el objeto sentencia.
        sentencia=(Statement)
conexion.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,Results
et.CONCUR_READ_ONLY);
    } catch (SQLException ex)
    {

Logger.getLogger(ProyectoServer.class.getName()).log(Level.SEVERE,
null, ex);
    }

}

    public void Cargar_Configuracion(){
        String IP="";
        String Ruta_Predeterminada="";

        try {
            /*Obtenemos la IP que tenemos almacenada como
predeterminada en la base de datos mediante sentencias SQL*/

            ResultSet r = sentencia.executeQuery("select * from
configuracion where ID='Computadora' ");
            r.beforeFirst();
            r.next();
            IP=r.getString("IP_Predeterminada");

Ruta_Predeterminada=r.getString("Ruta_Predeterminada");

            Ruta_Almacenamiento=Ruta_Predeterminada;

```

```

        txt_IP_Dispositivo.setText(IP);
        txt_Almacenaje.setText(Ruta_Predeterminada);

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error al
consultar la tabla trabajadores" + ex);
    }

}

/**
 * This method is called from within the constructor to
initialize the form.
 * WARNING: Do NOT modify this code. The content of this
method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {

    elegirFichero = new javax.swing.JFileChooser();
    elegirCarpeta = new javax.swing.JFileChooser();
    jPanel4 = new javax.swing.JPanel();
    jPanel3 = new javax.swing.JPanel();
    btn_examinar = new javax.swing.JButton();
    txt_ruta_archivo = new javax.swing.JTextField();
    btn_enviar = new javax.swing.JButton();
    jPanel2 = new javax.swing.JPanel();
    txt_Almacenaje = new javax.swing.JTextField();
    btn_Ruta_Almacenamiento = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();
    btn_Cambiar_IP = new javax.swing.JButton();
    txt_IP_PC = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();
    txt_IP_Dispositivo = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLO
SE);

    setBackground(new java.awt.Color(51, 51, 255));
    setForeground(new java.awt.Color(255, 146, 11));
    setIconImage(getIconImage());
    setResizable(false);

```

```

        jPanel4.setBackground(new java.awt.Color(178, 200,
198));
        jPanel4.setBorder(new javax.swing.border.LineBorder(new
java.awt.Color(178, 200, 198), 3, true));
        jPanel4.setForeground(new java.awt.Color(0, 153, 153));

        jPanel3.setBackground(new java.awt.Color(190, 186,
180));

jPanel3.setBorder(javax.swing.BorderFactory.createEtchedBorder()
);

        btn_examinar.setText("Examinar Archivo");
        btn_examinar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                btn_examinarActionPerformed(evt);
            }
        });

        txt_ruta_archivo.setEditable(false);
        txt_ruta_archivo.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                txt_ruta_archivoActionPerformed(evt);
            }
        });

        btn_enviar.setText("Enviar a dispositivo");
        btn_enviar.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                btn_enviarActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.LEADING)

        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(135, 135, 135)
            .addComponent(btn_examinar))

```



```

        .addGroup(jPanel3Layout.createSequentialGroup())
            .addGap(25, 25, 25)
            .addComponent(txt_ruta_archivo,
javax.swing.GroupLayout.PREFERRED_SIZE, 312,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(btn_enviar,
javax.swing.GroupLayout.PREFERRED_SIZE, 185,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(34, 34, 34))
    );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup())
            .addContainerGap(25, Short.MAX_VALUE)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.TRAILING)

        .addGroup(jPanel3Layout.createSequentialGroup())
            .addComponent(txt_ruta_archivo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
            .addComponent(btn_examinar))
            .addComponent(btn_enviar,
javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap())
    );

    jPanel2.setBackground(new java.awt.Color(190, 186,
180));

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder()
);

    txt_Almacenaje.setEditable(false);

    btn_Ruta_Almacenamiento.setText("Cambiar Ruta de
almacenamiento");
    btn_Ruta_Almacenamiento.addActionListener(new
java.awt.event.ActionListener() {
        public void
actionPerformed(java.awt.event.ActionEvent evt) {
            btn_Ruta_AlmacenamientoActionPerformed(evt);
        }
    });

```

```

        jLabel4.setText("Nota: El programa creará una subcarpeta
llamada \"NPECML\" en la ruta seleccionada");

        jLabel3.setText("Ruta de almacenamiento ->");

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(3, 3, 3)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.LEADING)
                        .addComponent(jLabel4)

.addGroup(jPanel2Layout.createSequentialGroup()
                                .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
                                    .addComponent(txt_Almacenaje,
javax.swing.GroupLayout.PREFERRED_SIZE, 206,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addGap(18, 18, 18)
                                        .addComponent(btn_Ruta_Almacenamiento)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                );
        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(14, 14, 14)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
                        .addComponent(jLabel3)
                        .addComponent(txt_Almacenaje,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(btn_Ruta_Almacenamiento))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
                                    .addComponent(jLabel4)
                                    .addGap(18, Short.MAX_VALUE)
                );

```

```

        jPanell1.setBackground(new java.awt.Color(190, 186,
180));

jPanell1.setBorder(javax.swing.BorderFactory.createEtchedBorder()
);

        btn_Cambiar_IP.setText("Establecer esta IP como
predeterminada");
        btn_Cambiar_IP.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                btn_Cambiar_IPActionPerformed(evt);
            }
        });

        txt_IP_PC.setEditable(false);

        jLabel1.setText("Introduce aqui la IP de tu dispositivo
movil ->");

        jLabel2.setText("Esta es la Dirección IP de su Ordenador
->");

        javax.swing.GroupLayout jPanell1Layout = new
javax.swing.GroupLayout(jPanell1);
        jPanell1.setLayout(jPanell1Layout);
        jPanell1Layout.setHorizontalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addGroup(jPanell1Layout.createSequentialGroup())
                .addContainerGap()

            .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.LEADING)
                .addComponent(jLabel1)
                .addComponent(jLabel2))
                .addGap(20, 20, 20)

            .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.LEADING)

            .addGroup(jPanell1Layout.createSequentialGroup())
                .addComponent(txt_IP_Dispositivo,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRE
LATED)
                .addComponent(btn_Cambiar_IP)
                .addComponent(txt_IP_PC,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(txt_IP_PC,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(12, 12, 12)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.BASELINE)
        .addComponent(txt_IP_Dispositivo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel1)
        .addComponent(btn_Cambiar_IP))
        .addGap(27, 27, 27))
    );

    jLabel5.setFont(new java.awt.Font("Arial Black", 0,
36)); // NOI18N
    jLabel5.setForeground(new java.awt.Color(0, 102, 102));
    jLabel5.setText("CASTLING");

    javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
    jPanel4.setLayout(jPanel4Layout);
    jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLay
out.Alignment.LEADING, false) .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addComponent(jPanel3,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap(19, Short.MAX_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup())

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jLabel5)
        .addGap(207, 207, 207))
    );
    jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup())
        .addComponent(jLabel5,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)
        .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELA
TED)
        .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addComponent(jPanel4,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
} // </editor-fold>

private void
btn_examinarActionPerformed(java.awt.event.ActionEvent evt) {
    /*Método para abrir el FileChooser y seleccionar un
    archivo, de dicho archivo se obtiene el nombre y la ruta y se
    almacena
    */
    int resp;

    //"elegirFichero" es el nombre del FileChooser
    resp = elegirFichero.showSaveDialog(this);

    if (resp == JFileChooser.APPROVE_OPTION) {
        JOptionPane.showMessageDialog(null,
        elegirFichero.getSelectedFile().toString());
        ruta = elegirFichero.getSelectedFile().toString();

Nombre_Archivo_Enviar=elegirFichero.getSelectedFile().getName().
toString();
        txt_ruta_archivo.setText(ruta);
        archivo_encontrado=true;
    } else if (resp == JFileChooser.CANCEL_OPTION) {
        JOptionPane.showMessageDialog(null, "Se pulsó la
opción Cancelar");
    }

}

private void
btn_enviarActionPerformed(java.awt.event.ActionEvent evt) {
    if (archivo_encontrado) {

/*Si hay un archivo seleccionado el método enviará el nombre del
archivo con su extensión a través del OutputStream del Socket*/

        IP_DISPOSITIVO=txt_IP_Dispositivo.getText();
        if (!IP_DISPOSITIVO.equals("")) {
            ObjectOutputStream oos;
            Socket s_enviar1;
            Socket s_enviar2;

```

```

        PrintWriter pw;

        boolean correcto=true;
        int contador = 0;
        int datos_entrada[];
        try {

            FileInputStream in = new
FileInputStream(ruta);
            long tamalo = in.getChannel().size();

            datos_entrada = new int[(int) tamalo];

            JOptionPane.showMessageDialog(null,
"Procediento a enviar "+ Nombre_Archivo_Enviar+" ...");

            s_enviar1 = new Socket(IP_DISPOSITIVO,
PUERTO_ENVIAR);
            pw=new
PrintWriter(s_enviar1.getOutputStream());
            pw.write(Nombre_Archivo_Enviar);
            pw.flush();
            pw.close();
            s_enviar1.close();

            /*Si el nombre del archivo se ha enviado correctamente significa
que la conexión entre el dispositivo móvil y el pc se ha
realizado correctamente, por lo que procedemos a dividir el
archivo en bytes y almacenarlos en un array de
enteros.*/

            while (true) {
                int byte_entrada = in.read();

                if (byte_entrada != -1) {

                    datos_entrada[contador] = byte_entrada;
                } else {
                    break;
                }
                contador++;
            }

            s_enviar2 = new Socket(IP_DISPOSITIVO,
PUERTO_ENVIAR);

            oos = new
ObjectOutputStream(s_enviar2.getOutputStream());
            Object Almacenamiento = datos_entrada;

            /*Procedemos a enviar el archivo convertido en Object a través
del Stream*/

```

```

        oos.writeObject(Almacenamiento);

        oos.flush();

        s_enviar2.close();
        in.close();
        oos.close();

        /*Si todo el proceso de envio se ha realizado correctamente
        procedemos a recoger la fecha y hora de nuestro computador para
        almacenar el registro de la transacción en nuestra base de datos
        */

        Calendar calendario =
        Calendar.getInstance();
        String hora, minutos, segundos,año,mes,dia;
        hora
        =String.valueOf(calendario.get(Calendar.HOUR_OF_DAY));
        minutos =
        String.valueOf(calendario.get(Calendar.MINUTE));
        segundos =
        String.valueOf(calendario.get(Calendar.SECOND));

        año=String.valueOf(calendario.get(Calendar.YEAR));

        mes=String.valueOf(calendario.get(Calendar.MONTH));

        dia=String.valueOf(calendario.get(Calendar.DAY_OF_MONTH));

        String fecha=año+"-"+mes+"-"+dia+"
        "+hora+": "+minutos+": "+segundos;

        String sql="insert into registro
        values('"+Nombre_Archivo_Enviar+"','"+fecha+"','Computadora')";

        sentencia.executeUpdate(sql);

        }catch(ConnectException e){
            JOptionPane.showMessageDialog(null, "Error
            al enviar el archivo\nComprueba la dirección IP del
            dispositivo");

            correcto = false;
        } catch (IOException e) {
            e.printStackTrace();
            correcto = false;
        } catch (SQLException ex) {

        Logger.getLogger(ProyectoServer.class.getName()).log(Level.SEVERE,
        null, ex);

            correcto = false;
        }

        if (correcto) {
            JOptionPane.showMessageDialog(null, "Archivo
            enviado con exito");
            archivo_encontrado = false;

```



```

        }else{
            JOptionPane.showMessageDialog(null, "Error
al enviar el archivo");
        }
    } else {
        JOptionPane.showMessageDialog(null, "Debe
introducir la direccion IP que aparece en su dispositivo para
conectarlo \nEjemplo: 192.168.0.54");
    }
    } else {
        JOptionPane.showMessageDialog(null, "Selecciona el
archivo a enviar primero");
    }
}

private void
btn_Cambiar_IPActionPerformed(java.awt.event.ActionEvent evt) {
    /*Este método coge el texto de la "IP del dispositivo" y
lo sustituye en la base de datos para establecer una nueva IP
predeterminada. Cuando el usuario abra la aplicación, esa IP
predeterminada aparecerá automáticamente permitiendo una
utilización msa sencilla y rápida*/

    String sql="";
    String IP_Nueva="";
    try {
        IP_Nueva=txt_IP_Dispositivo.getText();
        // TODO add your handling code here:
        sql="update configuracion set "
            + "IP_Predeterminada='"+IP_Nueva+"' "
            + "where ID='Computadora'";

        sentencia.executeUpdate(sql);
        IP_DISPOSITIVO=IP_Nueva;
        JOptionPane.showMessageDialog(null, "La IP
"+IP_DISPOSITIVO+" se ha seleccionado como predeterminada");

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error al
consultar la tabla configuracion" + ex);
    }

}

private void
btn_Ruta_AlmacenamientoActionPerformed(java.awt.event.ActionEven
t evt) {
    /*Método para elegir la carpeta donde se va a almacenar
los archivos recibidos. Este método utiliza una variación de un
FileChooser, siendo el objetivo de este directorios en lugar de
archivos, para poder almacenar las rutas de estos directorios
como nueva ruta de almacenamiento. Además, el método almacenará
esta nueva ruta en la base de datos como "ruta predeterminada"
    String sql="";

```

```

        String aux="";
        int resp;
        try {

elegirCarpeta.setFileSelectionMode(elegirCarpeta.DIRECTORIES_ONL
Y);

        elegirCarpeta.setAcceptAllFileFilterUsed(false);
        elegirCarpeta.setDialogTitle("Selecciona la
carpeta");
        resp = elegirCarpeta.showSaveDialog(this);

        if (resp == elegirCarpeta.APPROVE_OPTION) {

                aux =
elegirCarpeta.getCurrentDirectory().toString();
                String Ruta_Nueva=aux.replace("\\", "\\\"+\"\\");

                // TODO add your handling code here:
                sql = "update configuracion set "
                        + "Ruta_Predeterminada='" + Ruta_Nueva +
" , "
                        + "where ID='Computadora'";

                sentencia.executeUpdate(sql);

                Ruta_Almacenamiento = aux;

                txt_Almacenaje.setText(Ruta_Almacenamiento);

                JOptionPane.showMessageDialog(null, "Se ha
establecido una nueva ruta de almacenamiento predeterminada");

                } else if (resp == elegirCarpeta.CANCEL_OPTION) {
                        JOptionPane.showMessageDialog(null, "Se pulsó la
opción Cancelar");
                }
        } catch (SQLException ex) {

Logger.getLogger(ProyectoServer.class.getName()).log(Level.SEVERE,
null, ex);
        }

    }

    private void
txt_ruta_archivoActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */

```

```

        //<editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available,
stay with the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/p
laf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ProyectoServer.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ProyectoServer.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ProyectoServer.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
{

java.util.logging.Logger.getLogger(ProyectoServer.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ProyectoServer().setVisible(true);
                try {
                    ObtenerIP();
                    txt_IP_PC.setText(IP_SERVIDOR);
                } catch (Exception ex) {

Logger.getLogger(ProyectoServer.class.getName()).log(Level.SEVER
E, null, ex);
                }
            }
        });

        ObjectInputStream ois;

        FileOutputStream fichero_nuevo;
        Socket s_recibir1;

```

```

        Socket s_recibir2;
        ServerSocket ss;
        InputStreamReader isr;
        BufferedReader bf;
        String Nombre_Archivo = "default";
        try {

/*Accedemos a la información que hemos recibido por el InputStream del
Socket, en este primer recibo obtenemos el nombre y la extensión del
archivo enviado.
*/

                ss = new ServerSocket(PUERTO_RECIBIR);
                while (true) {
                        s_recibir1 = ss.accept();

                                isr = new
InputStreamReader(s_recibir1.getInputStream());
                                bf = new BufferedReader(isr);

                                Nombre_Archivo = bf.readLine();
                                s_recibir1.close();

                                /*Accedemos nuevamente al InputStream del Socket para recibir
una segunda tanda de datos los cuales contendrán el archivo en un
Array de bytes.
*/

                                s_recibir2 = ss.accept();

                                ois = new
ObjectInputStream(s_recibir2.getInputStream());

                                Object Almacenamiento = ois.readObject();

                                JOptionPane.showMessageDialog(null, "Archivo
"+Nombre_Archivo+" recibido con éxito, procediendo al
almacenamiento...");

/*Una vez tenemos este Array de bytes parseados y almacenados
procedemos a la creación (si no existe anteriormente) de una
subcarpeta que contendrá todos los archivos recibidos.
*/

                                File CarpetaProyecto = new
File(Ruta_Almacenamiento+"\\NPECML");

                                if (!CarpetaProyecto.exists()) {
                                        CarpetaProyecto.mkdir();
                                }

                                fichero_nuevo = new
FileOutputStream(CarpetaProyecto.getPath() + "\\\" +
Nombre_Archivo);

/*Una vez sepamos que la carpeta del almacenamiento está completa
procedemos a crear un archivo nuevo, cuyo nombre será el nombre que

```

hemos recibido en la primera tanda de datos y su contenido será los datos obtenidos en el Array de bytes que hemos obtenido en la segunda tanda
*/

```
        int auxiliar[] = (int[]) Almacenamiento;

        for (int i = 0; i < auxiliar.length; i++) {
            fichero_nuevo.write(auxiliar[i]);
        }

        JOptionPane.showMessageDialog(null, "Archivo
        "+Nombre_Archivo+" almacenado con exito");

        fichero_nuevo.close();

        //-----
        -----
        ois.close();
        isr.close();
        bf.close();
        s_recibir2.close();

    }
    } catch (IOException e) {
        e.printStackTrace();

    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }
}

// Variables declaration - do not modify
private javax.swing.JButton btn_Cambiar_IP;
private javax.swing.JButton btn_Ruta_Almacenamiento;
private javax.swing.JButton btn_enviar;
private javax.swing.JButton btn_examinar;
private javax.swing.JFileChooser elegirCarpeta;
private javax.swing.JFileChooser elegirFichero;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JTextField txt_Almacenaje;
private static javax.swing.JTextField txt_IP_Dispositivo;
private static javax.swing.JTextField txt_IP_PC;
private javax.swing.JTextField txt_ruta_archivo;
// End of variables declaration
}
```