



Instituto Politécnico Nacional.  
Escuela Superior De Cómputo.



Materia:  
Aplicaciones Para Comunicación En  
Red.

Tema:  
Tarea 1  
(Programa)

Profesor:  
Axel Ernesto Moreno Cervantes.

Alumno:  
Mario Alberto Miranda Sandoval.

Grupo:  
3CM5

**Modificar los programas Envia.java y Recibe.java proporcionados en clase, para el envío de múltiples archivos.**

Para realizar esta aplicación primero se genero una interfaz gráfica como la siguiente.



Para obtener mejor control de los archivos a enviar se generó una clase llamada **Archivo.java** donde se almacenará la información del archivo.

```
1. public class Archivo {
2.     public Archivo(String nombre, long tamaño, String path) {
3.         this.nombre = nombre;
4.         this.tamaño = tamaño;
5.         this.path = path;
6.     }
7.
8.     public String getNombre() { return nombre; }
9.     public long getTamaño() { return tamaño; }
10.    public String getPath() { return path; }
11.
12.    public void setNombre(String nombre) {
13.        this.nombre = nombre;
14.    }
15.
16.    public void setTamaño(long tamaño) {
17.        this.tamaño = tamaño;
18.    }
19.
20.    public void setPath(String path) {
21.        this.path = path;
22.    }
23.
24.    private String nombre;
```

```

25.     private long tamaño;
26.     private String path;
27. }

```

Se tienen dos opciones que es la de elegir archivo y una carpeta, usando un JFileChooser, para la segunda opción, lo que se hace es obtener los archivos dentro de la carpeta, añadirlos a la clase archivo y luego mandarlo.

A continuación, se incluye el código de la elección.

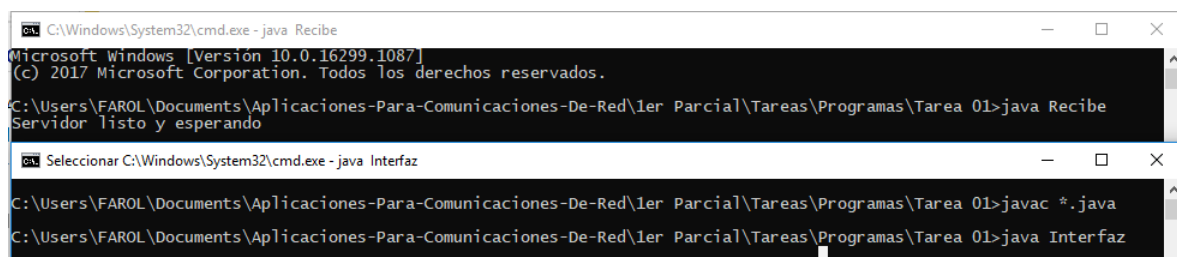
```

1. public void eleccionArchivo() {
2.     file = new JFileChooser();
3.     file.requestFocus();
4.     int r = file.showOpenDialog(Interfaz.this);
5.     if(r == JFileChooser.APPROVE_OPTION) {
6.         File f = file.getSelectedFile();
7.         misArchivos.add(new Archivo(f.getName(), f.length(), f.getAbsolutePath
8.     ));
9.         texto.append("\n" + f.getName() + "\t\t" + f.length());
10.    }
11.
12. public void carpetas(File carpeta, String destino) {
13.     if(destino.equalsIgnoreCase(""))
14.         destino = carpeta.getName();
15.     else
16.         destino = destino + "/" + carpeta.getName();
17.
18.     for(File f : carpeta.listFiles()) {
19.         if(f.isDirectory()) { carpetas(f, destino); }
20.         else {
21.             misArchivos.add(new Archivo(f.getName(), f.length(), f.getAbsolutePath
22.         ));
23.             texto.append("\n" + f.getName() + "\t\t" + f.length());
24.         }
25.     }

```

Para la parte de las carpetas el truco viene dado en la recursividad, aunque en una instancia se tenía contemplado el crear carpetas, al final se optó por pasar su contenido.

## Pruebas .



```

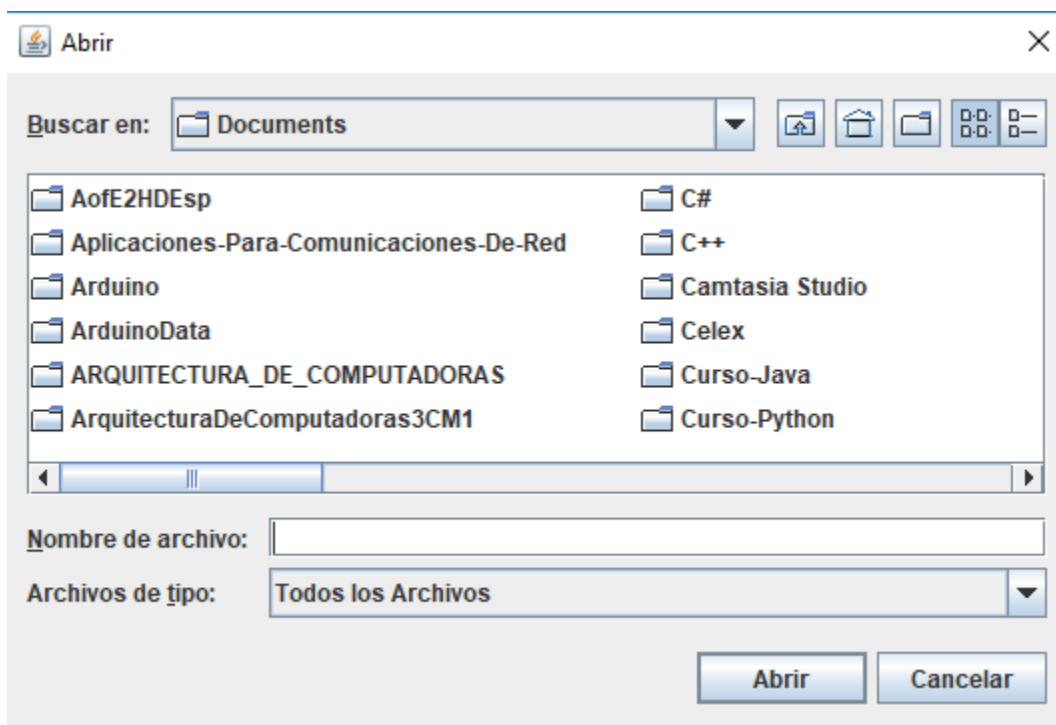
C:\Windows\System32\cmd.exe - java Recibe
Microsoft Windows [Version 10.0.16299.1087]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.
C:\Users\FAROL\Documents\Aplicaciones-Para-Comunicaciones-De-Red\1er Parcial\Tareas\Programas\Tarea 01>java Recibe
Servidor listo y esperando

C:\Windows\System32\cmd.exe - java Interfaz
C:\Users\FAROL\Documents\Aplicaciones-Para-Comunicaciones-De-Red\1er Parcial\Tareas\Programas\Tarea 01>javac *.java
C:\Users\FAROL\Documents\Aplicaciones-Para-Comunicaciones-De-Red\1er Parcial\Tareas\Programas\Tarea 01>java Interfaz

```



**Mandando Archivos.**





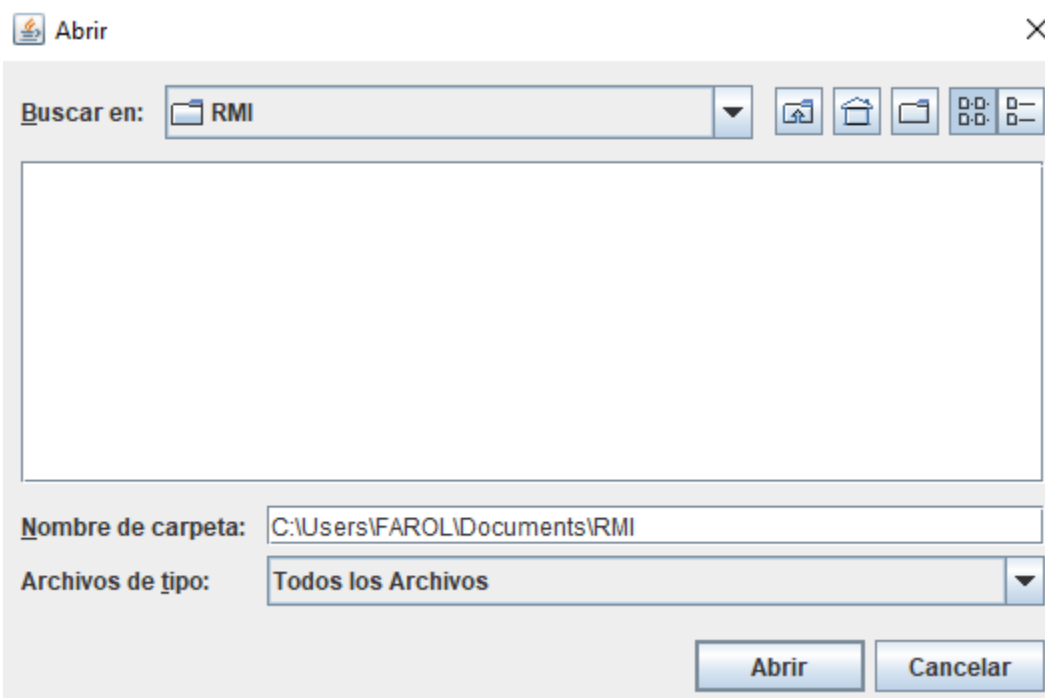
```
Prueba.txt,8,C:\Users\FAROL\Documents\Aplicaciones-Para-Comunicaciones-De-Red\Prueba.txt
Enviando el 100%
Archivo Enviado
Tabla-de-verdad-8bits.docx,17808,C:\Users\FAROL\Documents\ARQUITECTURA_DE_COMPUTADORAS\Tabla-de-verdad-8bits.docx
Enviando el 100%
Archivo Enviado
```

Este equipo > Documentos > Aplicaciones-Para-Comunicaciones-De-Red > 1er Parcial > Tareas > Programas > Tarea 01

Print Photo Print

	Nombre	Fecha de modifica...	Tipo	Tamaño
	Archivo.class	24/08/2019 06:40 ...	Archivo CLASS	1 KB
	Archivo	21/08/2019 08:45 a...	Archivo JAVA	1 KB
	Interfaz\$1.class	24/08/2019 06:40 ...	Archivo CLASS	1 KB
	Interfaz\$2.class	24/08/2019 06:40 ...	Archivo CLASS	1 KB
	Interfaz\$3.class	24/08/2019 06:40 ...	Archivo CLASS	2 KB
	Interfaz\$4.class	22/08/2019 05:38 ...	Archivo CLASS	2 KB
	Interfaz.class	24/08/2019 06:40 ...	Archivo CLASS	6 KB
	Interfaz	22/08/2019 05:38 ...	Archivo JAVA	6 KB
	Prueba	19/08/2019 07:54 a...	Documento de tex...	1 KB
	Recibe.class	24/08/2019 06:40 ...	Archivo CLASS	2 KB
	Recibe	22/08/2019 05:38 ...	Archivo JAVA	2 KB
	Tabla-de-verdad-8bits	15/02/2019 02:33 a...	Documento de Mi...	18 KB

Seleccionando una carpeta.



Este equipo > Documentos > Aplicaciones-Para-Comunicaciones-De-Red > 1er Parcial > Tareas > Programas > Tarea 01

Print Photo Print

Nombre	Fecha de modifica...	Tipo	Tamaño
Archivo.class	24/08/2019 06:40 ...	Archivo CLASS	1 KB
Archivo	21/08/2019 08:45 a...	Archivo JAVA	1 KB
Cliente.class	04/06/2018 11:16 ...	Archivo CLASS	2 KB
Cliente	04/06/2018 11:16 ...	Archivo JAVA	2 KB
datos	06/05/2018 07:27 ...	Documento de tex...	1 KB
Interfaz\$1.class	24/08/2019 06:40 ...	Archivo CLASS	1 KB
Interfaz\$2.class	24/08/2019 06:40 ...	Archivo CLASS	1 KB
Interfaz\$3.class	24/08/2019 06:40 ...	Archivo CLASS	2 KB
Interfaz\$4.class	22/08/2019 05:38 ...	Archivo CLASS	2 KB
Interfaz.class	24/08/2019 06:40 ...	Archivo CLASS	6 KB
Interfaz	22/08/2019 05:38 ...	Archivo JAVA	6 KB
IPPregunta.class	04/06/2018 11:16 ...	Archivo CLASS	1 KB
IPPregunta	04/06/2018 10:17 ...	Archivo JAVA	1 KB
no.policy	08/10/2010 08:37 a...	Archivo POLICY	1 KB
ObjetoRemoto.class	04/06/2018 11:16 ...	Archivo CLASS	2 KB
ObjetoRemoto	04/06/2018 11:05 ...	Archivo JAVA	2 KB
Prueba	19/08/2019 07:54 a...	Documento de tex...	1 KB
Recibe.class	24/08/2019 06:40 ...	Archivo CLASS	2 KB
Recibe	22/08/2019 05:38 ...	Archivo JAVA	2 KB
Tabla-de-verdad-8bits	15/02/2019 02:33 a...	Documento de Mi...	18 KB

Como se puede observar todos los archivos fueron mandados con éxito.

Un problema que surgió al generar esta tarea, fue el mal uso del socket, el cual se estaba escribiendo demasiada información en él, hasta que se le pregunto al profesor acerca de este y su uso al fin, se opto por crear y cerrar un socket cada vez que se mandaba un archivo.

## Código

### Interfaz.java

```

1. import javax.swing.*;
2. import java.awt.event.*;
3. import java.awt.*;
4. import java.net.*;
5. import java.io.*;
6. import java.util.*;
7.
8. public class Interfaz extends JFrame {
9.     private static final long serialVersionUID = 1L;
10.    public Interfaz() {
11.        setBounds(450, 150, 500, 300);
12.        setTitle("Envio de archivos");
13.        setResizable(false);
14.
15.        panelPrincipal = new JPanel();
16.        panelPrincipal.setLayout(new BorderLayout());

```

```

17.     panelSuperior = new JPanel();
18.     panelInferior = new JPanel();
19.     elegirArchivo = new JButton("Elegir archivo");
20.     enviarArchivo = new JButton("Enviar");
21.     enviarCarpeta = new JButton("Enviar Carpeta");
22.     //enviarCarpeta.setEnabled(false);
23.     //conectar = new JButton("Conectar");
24.     estado = new JLabel("Estado: Desconectado");
25.     porcentajeE = new JLabel("Porcentaje: ");
26.     texto = new JTextArea(50, 100);
27.     texto.setEditable(false);
28.     texto.append("Archivo:\\t\\tTama\\u00F1o:");
29.     misArchivos = new ArrayList<>();
30.
31.
32.     elegirArchivo.addActionListener(new ActionListener(){
33.         public void actionPerformed(ActionEvent arg0) {
34.             eleccionArchivo();
35.         }
36.     });
37.
38.     enviarArchivo.addActionListener(new ActionListener() {
39.         public void actionPerformed(ActionEvent e) {
40.             enviarArchivos();
41.         }
42.     });
43.
44.     enviarCarpeta.addActionListener(new ActionListener() {
45.         public void actionPerformed(ActionEvent e) {
46.             directory = new JFileChooser();
47.             directory.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
48.             directory.requestFocus();
49.             int r = directory.showOpenDialog(Interfaz.this);
50.             if (r == JFileChooser.APPROVE_OPTION) {
51.                 carpetas(directory.getSelectedFile(), "" + directory.getCurren
tDirectory());
52.             }
53.         }
54.     });
55.
56.     panelSuperior.setLayout(new BorderLayout());
57.     panelSuperior.add(estado, BorderLayout.NORTH);
58.     panelSuperior.add(porcentajeE, BorderLayout.CENTER);
59.     panelInferior.add(elegirArchivo);
60.     panelInferior.add(enviarArchivo);
61.     panelInferior.add(enviarCarpeta);
62.     //panelInferior.add(conectar);
63.     panelPrincipal.add(panelSuperior, BorderLayout.NORTH);
64.     panelPrincipal.add(texto, BorderLayout.CENTER);
65.     panelPrincipal.add(panelInferior, BorderLayout.SOUTH);
66.     add(panelPrincipal);
67.
68.
69.     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
70.     setVisible(true);
71. }
72.
73. public void eleccionArchivo() {
74.     file = new JFileChooser();
75.     file.requestFocus();
76.     int r = file.showOpenDialog(Interfaz.this);

```



```

77.         if(r == JFileChooser.APPROVE_OPTION) {
78.             File f = file.getSelectedFile();
79.             misArchivos.add(new Archivo(f.getName(), f.length(), f.getAbsolutePath
    ()));
80.             texto.append("\n" + f.getName() + "\t\t" + f.length());
81.         }
82.     }
83.
84.     public void enviarArchivos() {
85.         enviarArchivo.setEnabled(false);
86.         try {
87.             for(Archivo a : misArchivos) {
88.                 cl = new Socket(HOST, PUERTO);
89.                 estado.setText("Estado: Conectado");
90.                 dos = new DataOutputStream(cl.getOutputStream());
91.                 System.out.println(a.getNombre() + "," + a.getTamanio() + "," + a.
getPath());
92.                 long e = 0;
93.                 int n = 0;
94.                 int porcentaje = 0;
95.                 dis = new DataInputStream(new FileInputStream(a.getPath()));
96.                 dos.writeUTF(a.getNombre());
97.                 dos.flush();
98.                 dos.writeLong(a.getTamanio());
99.                 dos.flush();
100.                 while (e < a.getTamanio()) {
101.                     byte[] b = new byte[2000];
102.                     n = dis.read(b);
103.                     e += n;
104.                     dos.write(b, 0, n);
105.                     dos.flush();
106.                     porcentaje = (int) ((e*100)/a.getTamanio());
107.                     System.out.print("\rEnviando el " + porcentaje + "
    %");
108.                     porcentajeE.setText("Porcentaje: " + porcentaje +
    "%");
109.                 }
110.                 System.out.println("\nArchivo Enviado");
111.                 dis.close();
112.                 dos.close();
113.                 cl.close();
114.             }
115.             misArchivos.clear();
116.             estado.setText("Estado: Desconectado");
117.             texto.setText("");
118.         } catch (Exception e1) {
119.             estado.setText("Estado: Desconectado");
120.             e1.printStackTrace();
121.         }
122.     }
123.
124.     public void carpetas(File carpeta, String destino) {
125.         if(destino.equalsIgnoreCase(""))
126.             destino = carpeta.getName();
127.         else
128.             destino = destino + "/" + carpeta.getName();
129.
130.         for(File f : carpeta.listFiles()) {
131.             if(f.isDirectory()) { carpetas(f, destino); }
132.             else {

```

```

133.                misArchivos.add(new Archivo(f.getName(), f.length(), f.get
    AbsolutePath()));
134.                texto.append("\n" + f.getName() + "\t\t" + f.length());
135.            }
136.        }
137.    }
138.
139.    public static void main(String[] args) {
140.        new Interfaz();
141.    }
142.
143.    private JPanel panelPrincipal;
144.    private JPanel panelSuperior;
145.    private JPanel panelInferior;
146.    private JButton elegirArchivo, enviarArchivo;
147.    private JButton enviarCarpeta;
148.    private JButton conectar;
149.    private JLabel estado, porcentajeE;
150.    private JTextArea texto;
151.    final int PUERTO = 9000;
152.    final String HOST = "127.0.0.1";
153.    private Socket cl;
154.    private JFileChooser file;
155.    private JFileChooser directory;
156.    private ArrayList <Archivo> misArchivos;
157.    private DataOutputStream dos;
158.    private DataInputStream dis;
159.    }

```

## Recibe.java

```

1. import java.net.*;
2. import java.io.*;
3.
4. public class Recibe {
5.     public static void main(String[] args) {
6.         try {
7.             final int PUERTO = 9000;
8.             ServerSocket s = new ServerSocket(PUERTO);
9.             System.out.println("Servidor listo y esperando");
10.            for(;;) {
11.                Socket cl = s.accept();
12.                DataInputStream dis = new DataInputStream(cl.getInputStream());
13.                DataOutputStream dos;
14.
15.                String nombre = dis.readUTF();
16.                long tam = dis.readLong();
17.                System.out.println("Preparado para recibir el archivo: " + cl.
getInetAddress() + ":" + cl.getPort());
18.
19.                long r = 0;
20.                int n = 0;
21.                int porcentaje = 0;
22.                dos = new DataOutputStream(new FileOutputStream(nombre));
23.
24.                while(r < tam) {
25.                    byte[] b = new byte[2000];
26.                    n = dis.read(b);
27.                    r += n;
28.                    dos.write(b, 0, n);

```

```

29.             dos.flush();
30.             porcentaje = (int) ((r*100) / tam);
31.             System.out.println("Se ha recibido " + porcentaje + "%");

32.         }
33.         dos.close();
34.         System.out.println("Se completo la descarga");
35.         dis.close();
36.         cl.close();
37.     }
38. } catch(Exception e) {
39.     e.printStackTrace();
40. }
41. }
42. }

```

## Archivo.java

```

1. public class Archivo {
2.     public Archivo(String nombre, long tamaño, String path) {
3.         this.nombre = nombre;
4.         this.tamaño = tamaño;
5.         this.path = path;
6.     }
7.
8.     public String getNombre() { return nombre; }
9.     public long getTamaño() { return tamaño; }
10.    public String getPath() { return path; }
11.
12.    public void setNombre(String nombre) {
13.        this.nombre = nombre;
14.    }
15.
16.    public void setTamaño(long tamaño) {
17.        this.tamaño = tamaño;
18.    }
19.
20.    public void setPath(String path) {
21.        this.path = path;
22.    }
23.
24.    private String nombre;
25.    private long tamaño;
26.    private String path;
27. }

```

