

Explicación

La práctica básicamente se basó en modificar el cliente y servidor del profesor para el envío de un solo archivo y la inclusión del algoritmo de Neagle, de los cuales mostraremos los cambios:

Servidor

```
/* Se recibió el tamaño del buffer que modificó el cliente en la variable int buffer */
DataInputStream dis = new DataInputStream(cl.getInputStream());
int buffer = dis.readInt();

/*Se estableció el valor de envío y recepción del buffer de acuerdo con la línea anterior */
cl.setSendBufferSize(buffer);
cl.setReceiveBufferSize(buffer);
cl.setTcpNoDelay(dis.readBoolean());

System.out.println("El tamaño del buffer de envío de paquetes es:" + cl.getSendBufferSize());
System.out.println("El tamaño del buffer para recibir paquetes es:" + cl.getReceiveBufferSize());

/*A partir de las siguientes líneas se añadió todo en un bucle y las variables que en el código ejemplo solo
almacenaban un solo dato se convirtieron en arreglos ese fue el principal cambio que se ejecutó a partir del
código dado */

int archivos = dis.readInt();
String[] nombres = new String[archivos]; //Nombres
long[] tam = new long[archivos]; //Tamaño

for (int i = 0; i < archivos; i++) {
```

```
nombres[i] = dis.readUTF();
tam[i] = dis.readLong();
}

int i = 0;
while (i < archivos) {
    byte[] b = new byte[buffer];
    DataOutputStream dos = new DataOutputStream(new FileOutputStream(nombres[i]));
    long recibidos = 0;
    int n, porcentaje;
    while (recibidos < tam[i]) {
        n = dis.readInt();
        n = dis.read(b, 0, n);
        dos.write(b, 0, n);
        recibidos = recibidos + n;
        dos.flush();
        porcentaje = (int) (recibidos * 100 / tam[i]);
        System.out.print("\nRecibido: " + porcentaje + "%\r");
    } // While
    System.out.println("\nRecibimos el archivo:" + nombres[i] + " recibido");
    dos.close();
    i++;
    Thread.sleep(100); // Se añadio un sleep por lo rápido que se ejecuta la recepción de archivos.
}

dis.close();
cl.close();
}

    } catch (Exception e) {
        e.printStackTrace();
    } // catch
}
}
```

Cliente

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
Socket cl = new Socket("localhost", 1234);

        /* Se preguntó al usuario el tamaño del buffer que desea usar */
String respuesta;
System.out.println("Ingrese el tamaño del buffer: ");
respuesta = br.readLine();
int buffer = Integer.parseInt(respuesta);

        /* Se estableció el tamaño del buffer de acuerdo con el usuario */
cl.setSendBufferSize(buffer);
cl.setReceiveBufferSize(buffer);

System.out.println("El tamaño del buffer de envió de paquetes es : " + cl.getSendBufferSize());
System.out.println("El tamaño del buffer para recibir paquetes es : " + cl.getReceiveBufferSize());

        /* Se preguntó al usuario si deseada utilizar el algoritmo de nagle */

System.out.println("¿Desea activar el algoritmo de nagle? (s/n)");
respuesta = br.readLine();

DataOutputStream dos = new DataOutputStream(cl.getOutputStream());

dos.writeInt(buffer);
dos.flush();

        /* Se activa el algoritmo de nagle de acuerdo con la decisión del usuario */

if ("si".equalsIgnoreCase(respuesta) || "s".equalsIgnoreCase(respuesta)) {
    cl.setTcpNoDelay(true);
    dos.writeBoolean(true);
} else {
```

```
cl.setTcpNoDelay(false);
dos.writeBoolean(false);
}
dos.flush();

System.out.println("¿Algoritmo de nagle activado? " + cl.getTcpNoDelay());

/* Se cambia setSelectionEnabled por SetMultiSelectionEnabled para que pueda seleccionar varios archivos*/

JFileChooser jf = new JFileChooser(".");
jf.setMultiSelectionEnabled(true);

int r = jf.showOpenDialog(null);
if (r == JFileChooser.APPROVE_OPTION) {
    /* De la misma forma se cambia la opción de SelectFile para múltiples archivos*/

    File[] f = jf.getSelectedFiles(); //Manejador

    dos.writeInt(f.length);
    dos.flush();

    /*A partir de las siguientes líneas se añadió todo en un bucle y las variables que en el código ejemplo solo
    almacenaban un solo dato se convirtieron en arreglos ese fue el principal cambio que se ejecutó a partir del
    código dado */

    String[] direcciones = new String[f.length]; //Direcciones
    String[] nombres = new String[f.length]; //Nombres
    long[] tam = new long[f.length]; //Tamaño

    for (int i = 0; i < f.length; i++) {
        nombres[i] = f[i].getName();
        tam[i] = f[i].length();
        direcciones[i] = f[i].getAbsolutePath();
        dos.writeUTF(f[i].getName());
        dos.flush();
        dos.writeLong(f[i].length());
    }
}
```

```
        dos.flush();
    }

    for (int i = 0; i < f.length; i++) {
        DataInputStream dis = new DataInputStream(new FileInputStream(direcciones[i]));
        System.out.println("Enviando archivo: " + nombres[i]);
        byte[] b = new byte[buffer];
        long enviados = 0;
        int porcentaje, n;

        while (enviados < tam[i] ) {
            n = dis.read(b);
            dos.writeInt(n);
            dos.flush();
            dos.write(b, 0, n);
            enviados = enviados + n;
            dos.flush();
            porcentaje = (int) (enviados * 100 / tam[i]);
            System.out.print("\nEnviado: " + porcentaje + "%\r");
        } //While
        System.out.println("\n\nArchivo enviado" + nombres[i] + " enviado");
        dis.close();
    } //fin del for
    dos.close();
    cl.close();
} //if
} catch (IOException e) {
    e.printStackTrace();
}
}
```

/ Pasaremos a ejecutar el código en este reporte con capturas de pantalla */*

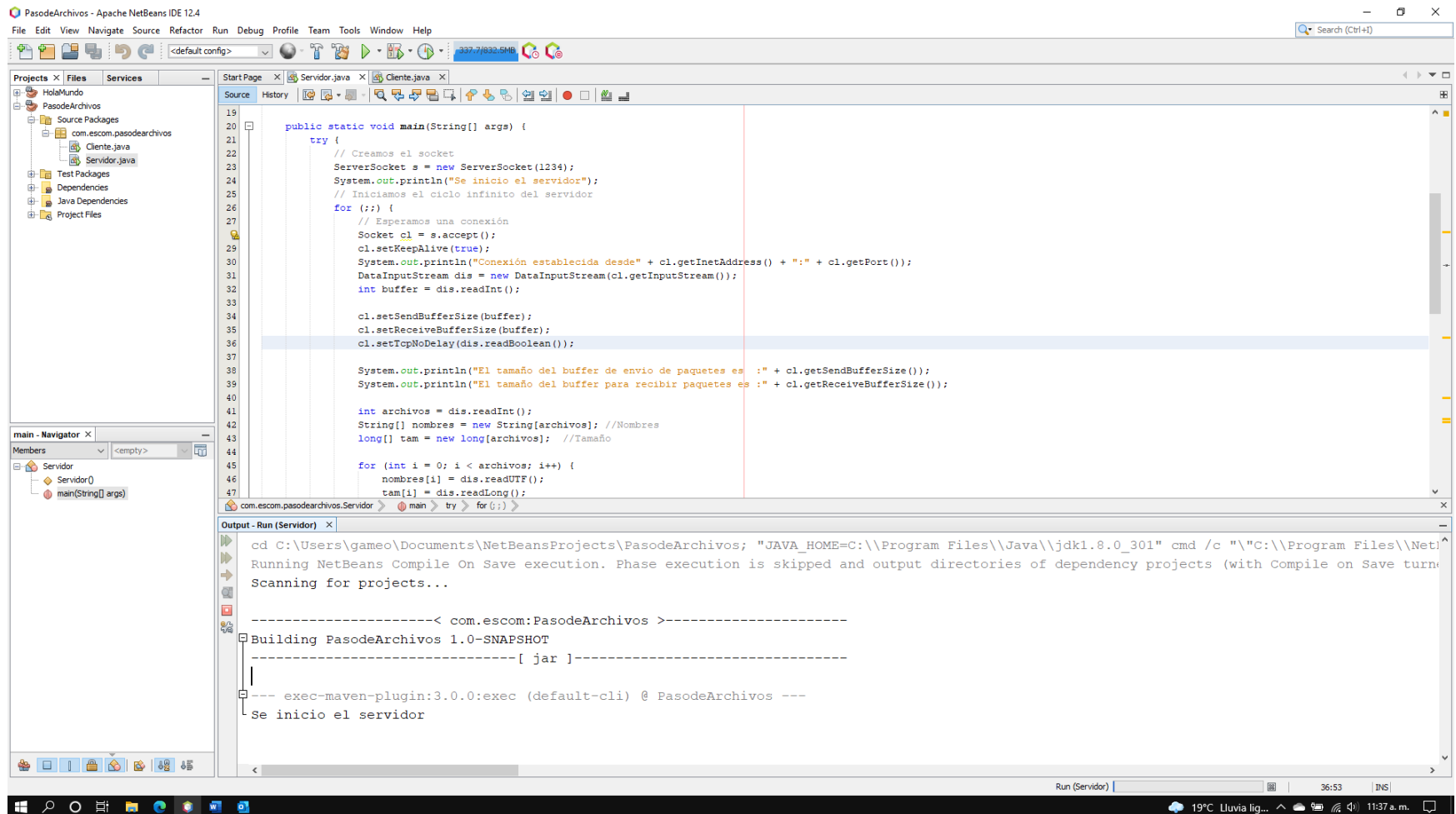


Ilustración 1. Se muestra el servidor iniciado

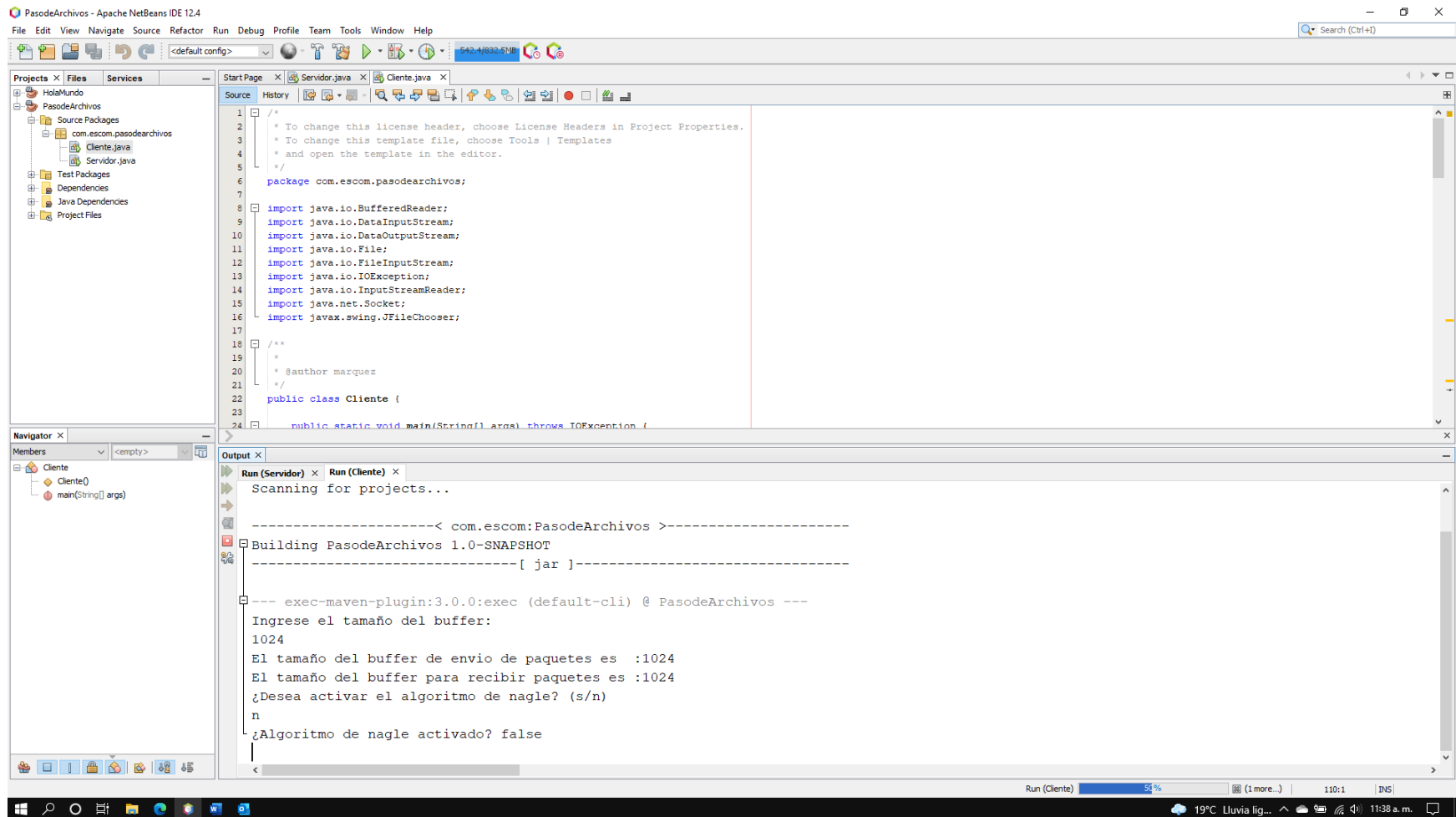


Ilustración 2. Se muestra las preguntas al usuario sobre el tamaño del buffer y algoritmo de nagle

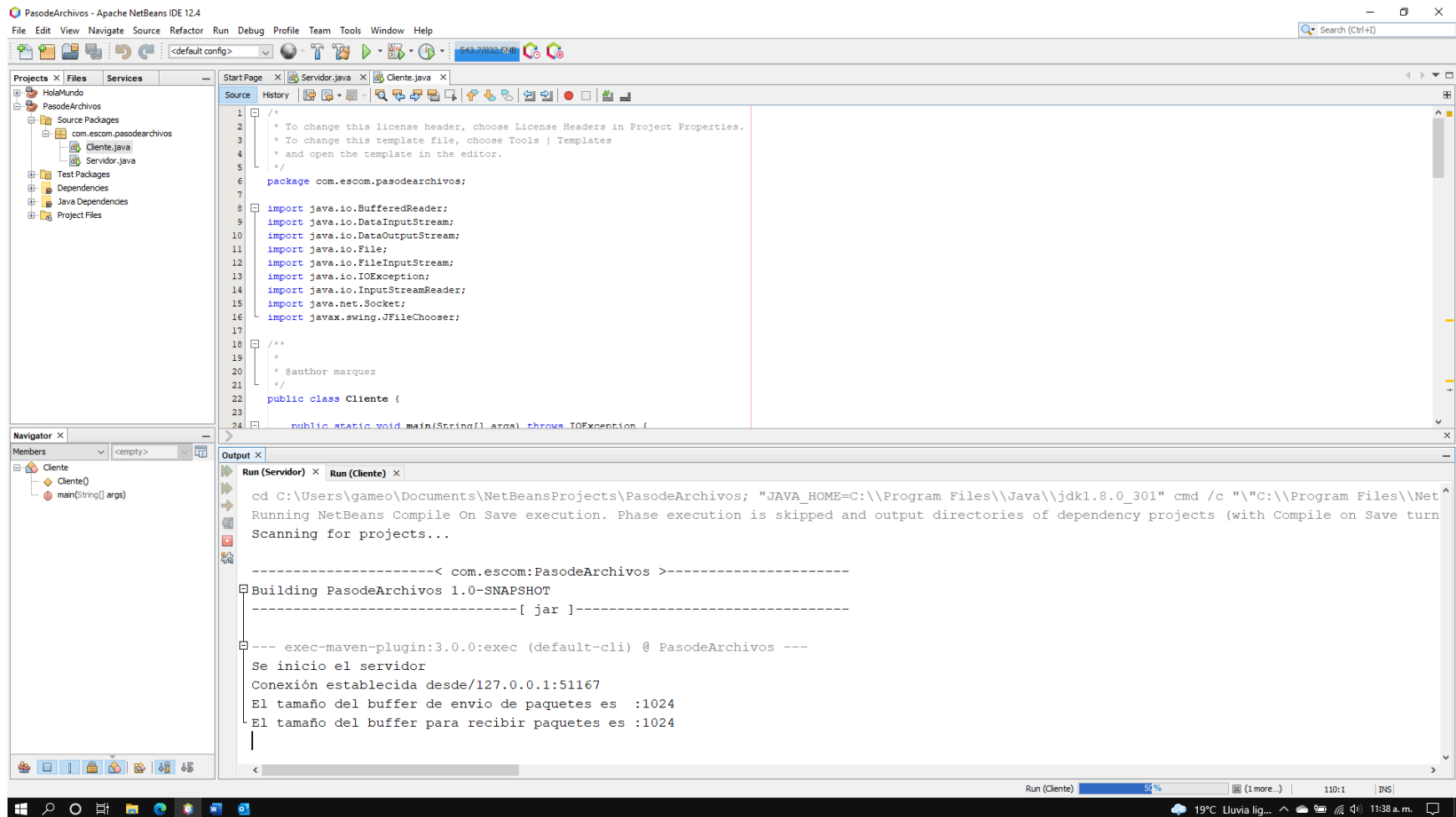


Ilustración 3. Se muestra la recepción del tamaño del buffer establecido por el usuario

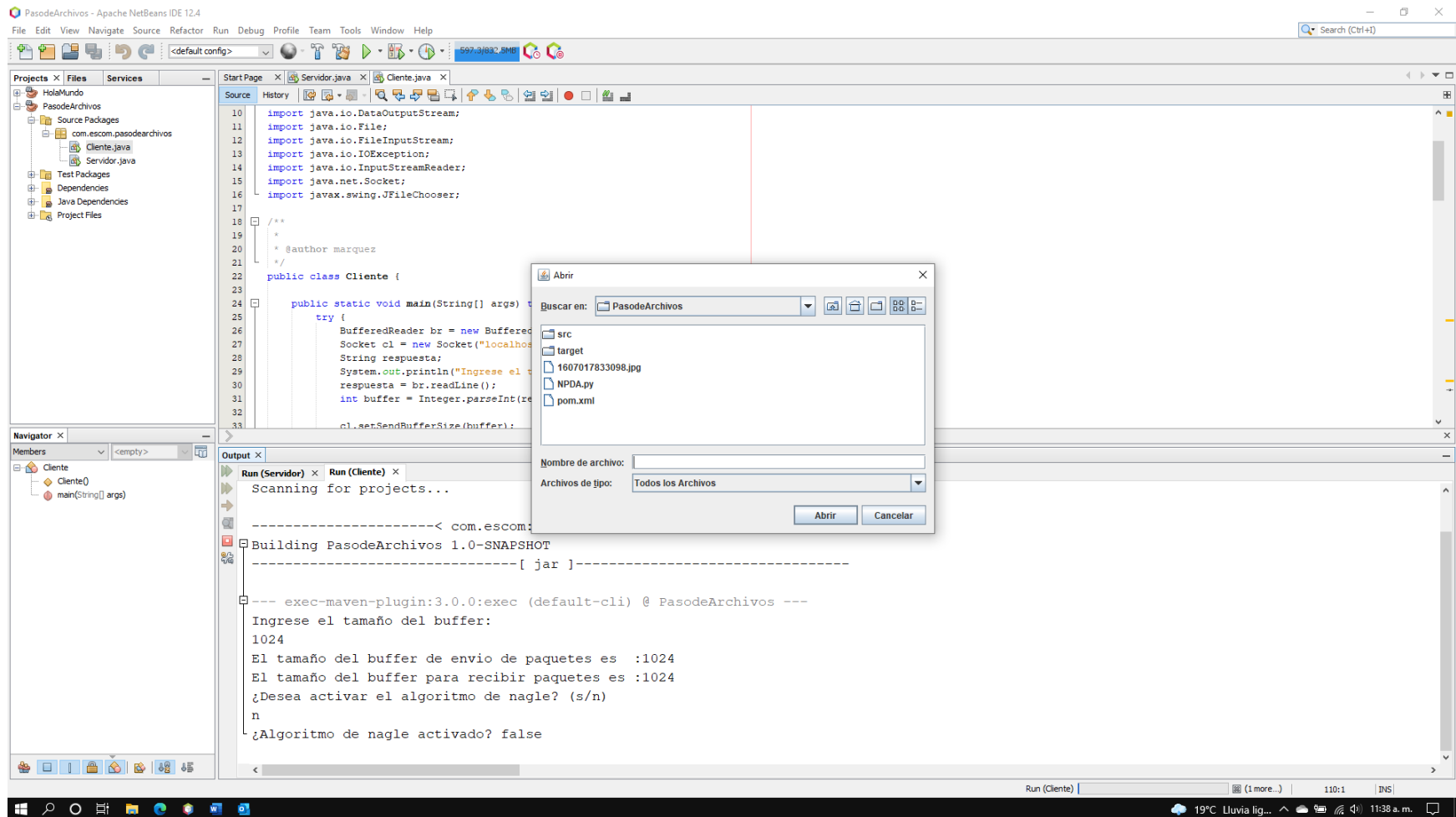


Ilustración 4. Se muestra un menú visual en donde encontramos el proyecto actual, los archivos que se visualizan ahí son de pruebas anteriores.

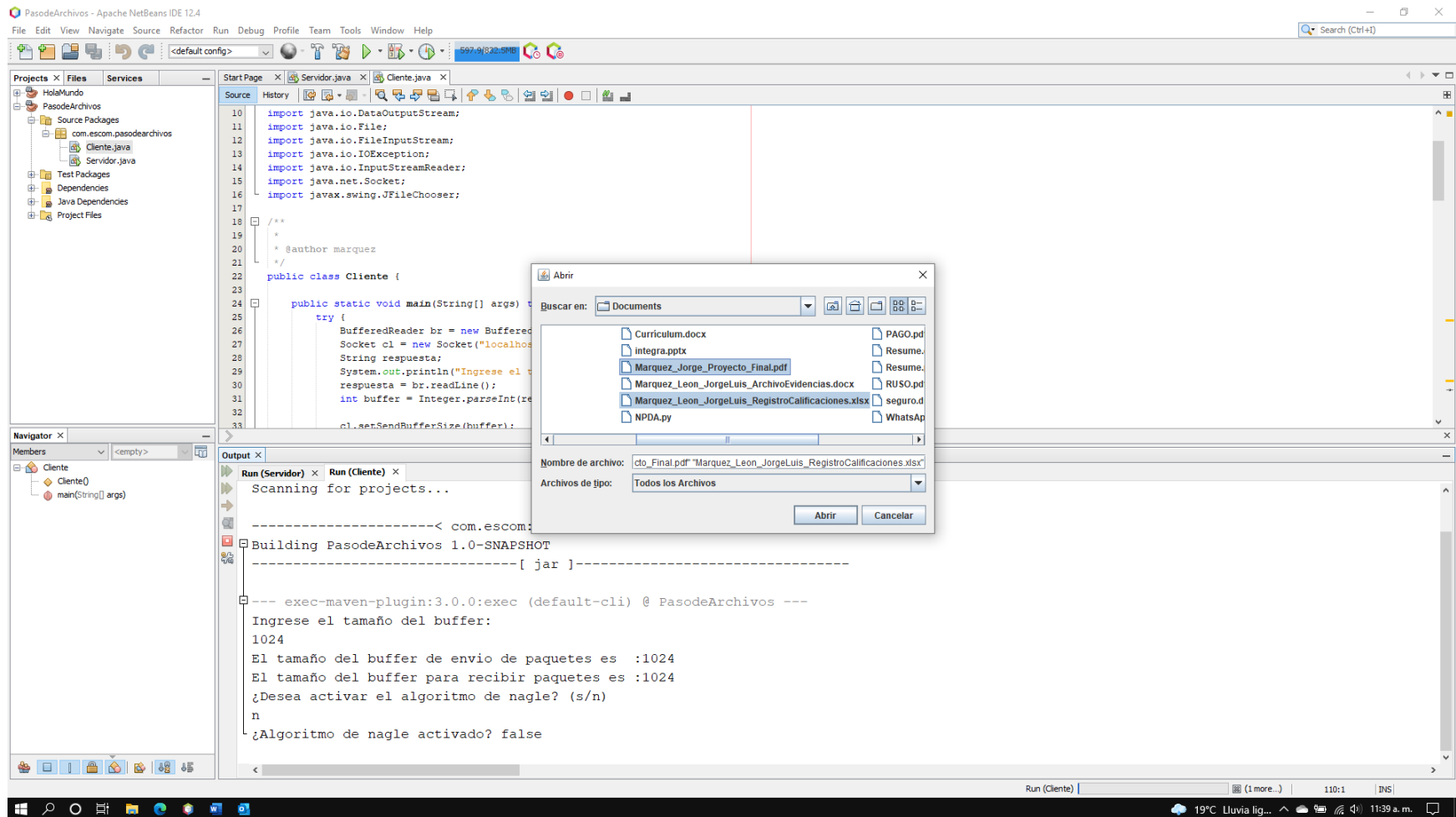


Ilustración 5. Se seleccionan los nuevos archivos a enviar.

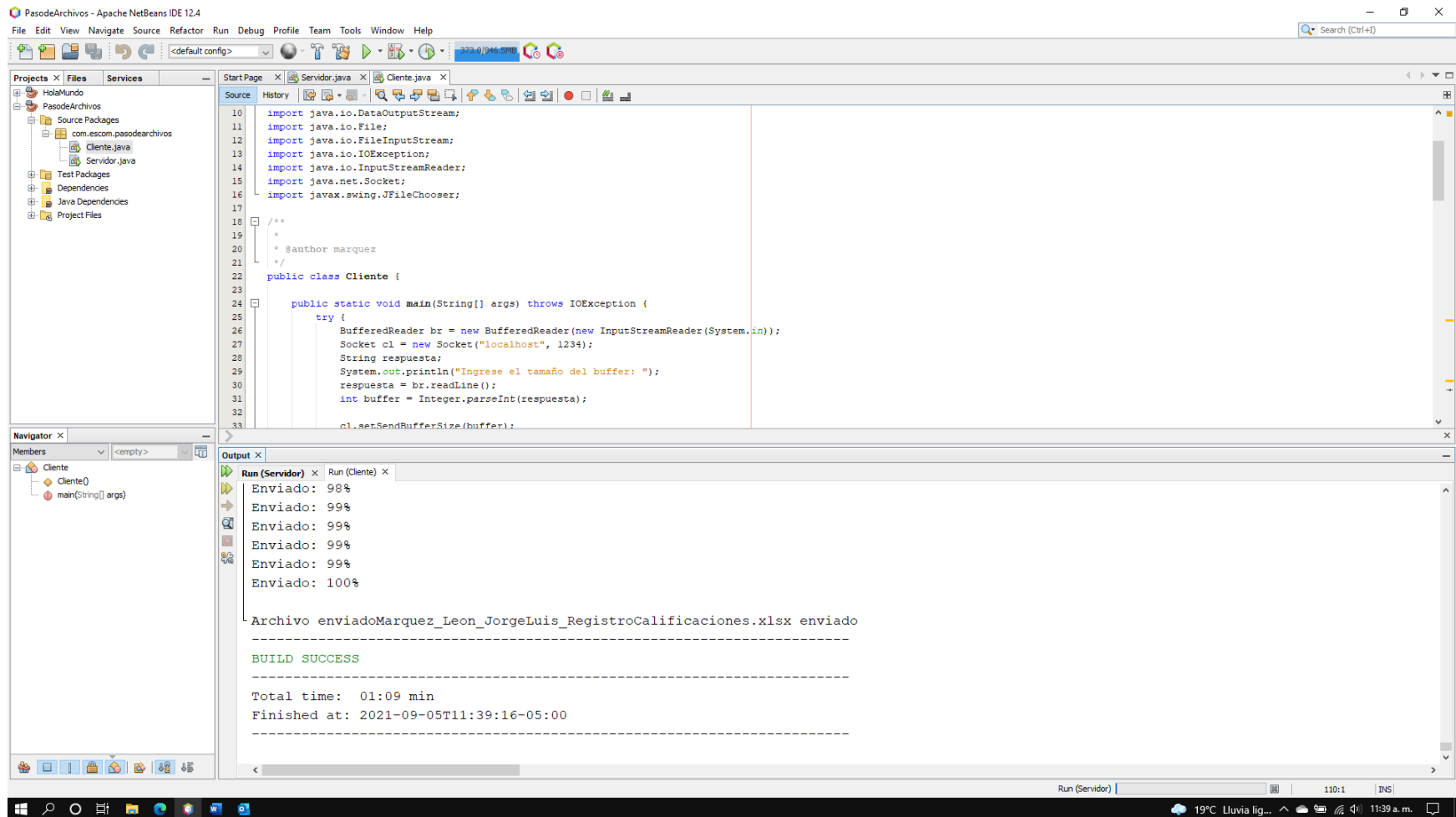


Ilustración 6. Se muestra el cliente envió los dos archivos

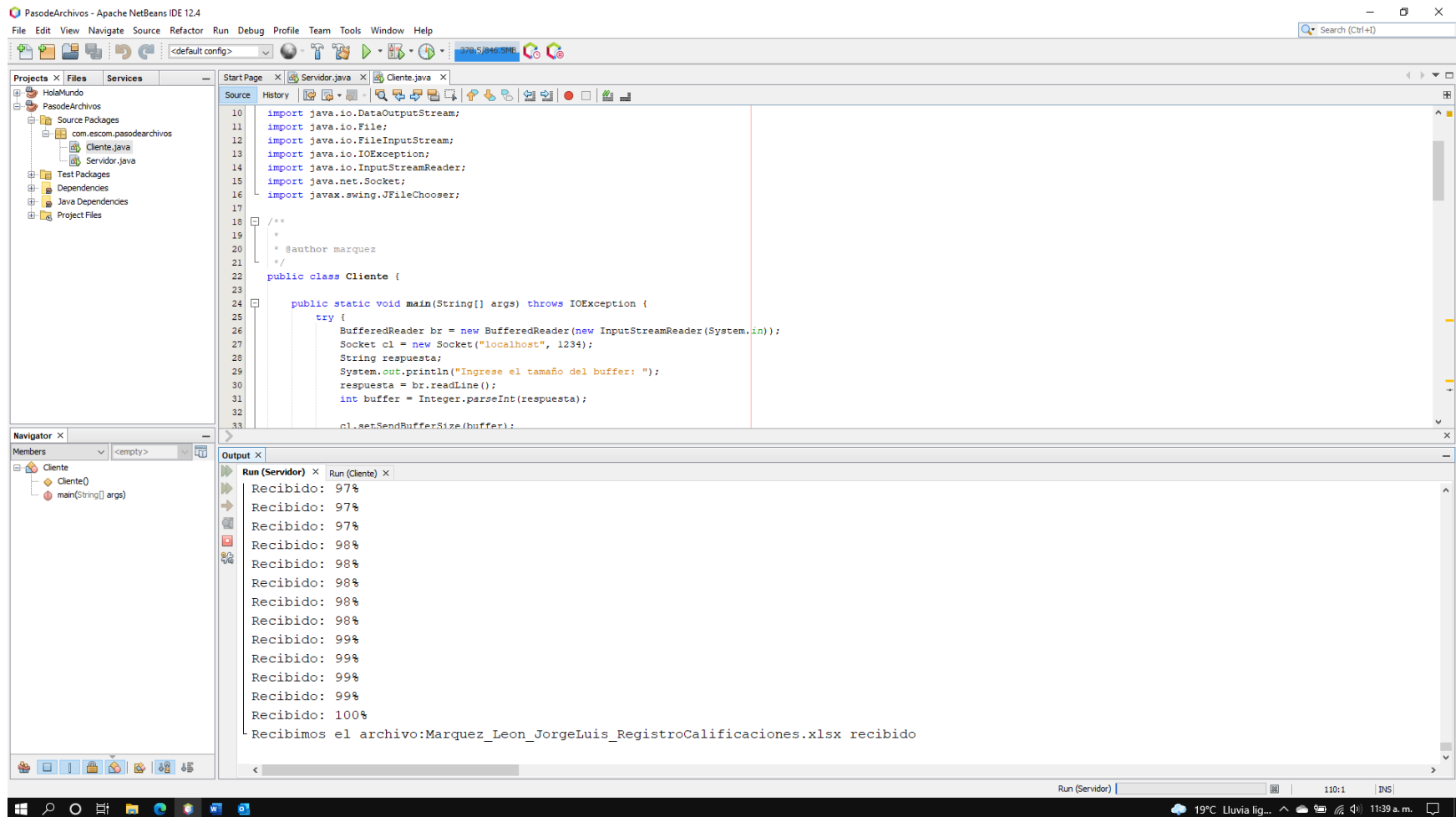


Ilustración 7. Se muestra que el servidor recibió los archivos.

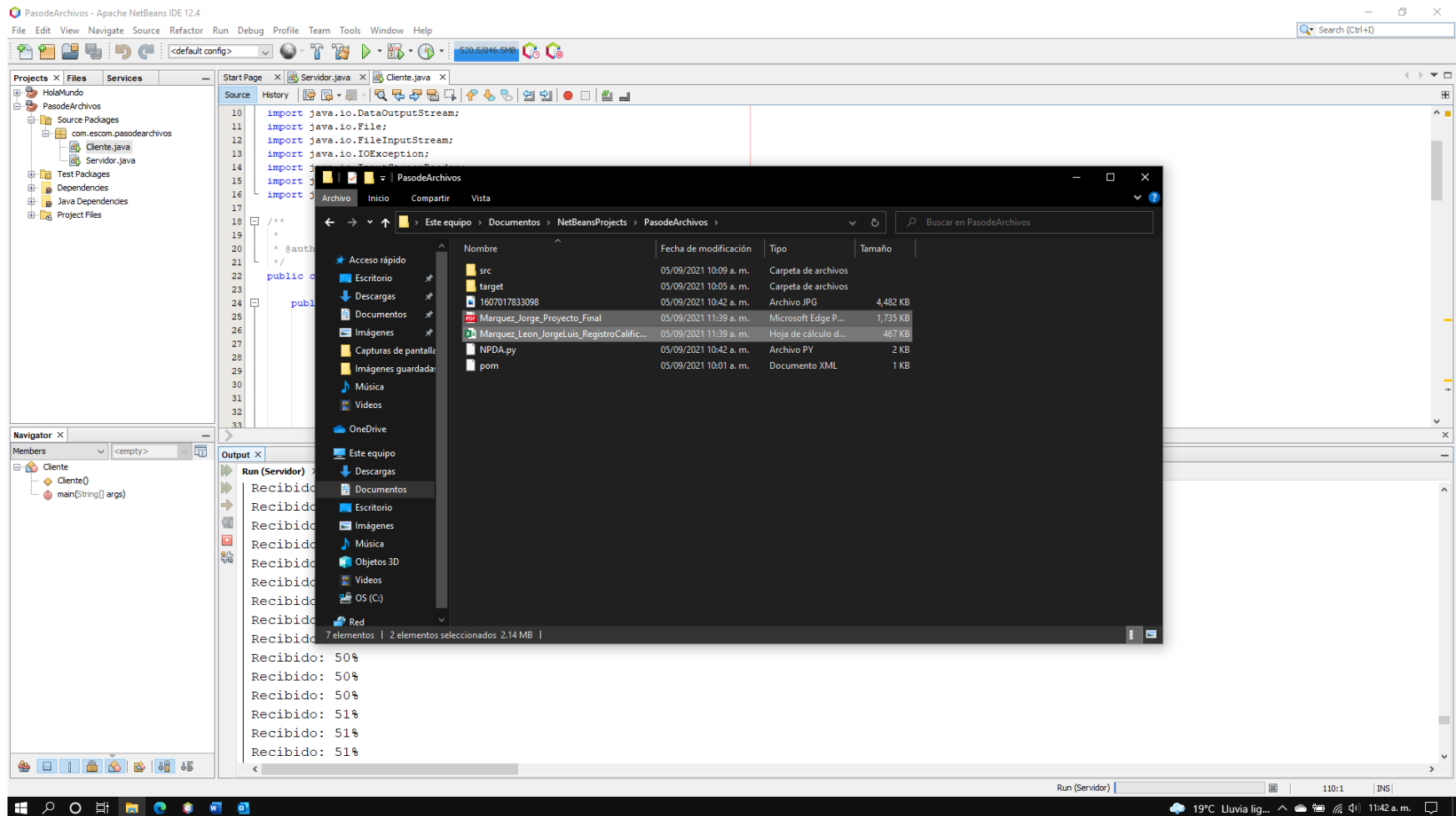


Ilustración 8. Se muestran los archivos recibidos en la carpeta del proyecto.