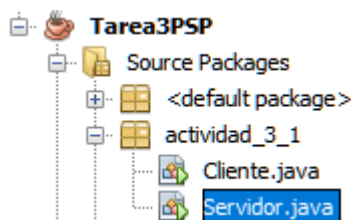


### ACTIVIDAD 3.1

Se nos pide que creamos una aplicación del tipo cliente-servidor en la que el servidor genere un número aleatorio de 0-100 y el cliente tenga que adivinar el número que ha generado lanzando al servidor números. Y este como ayuda le irá diciendo si es mayor, menor o si ha acertado y es el número correcto.

Para diseñar esta tarea lo primero que vamos a hacer es crear 2 clases dentro del proyecto: Servidor y Cliente



- Clase Servidor: En esta clase vamos a crear el ServerSocket con el puerto 2000 que se quedará a la escucha de recibir la petición de cliente y en el momento de recibirla se crea el socket cliente. De esta manera ya estarían conectadas las 2 partes de nuestra aplicación.

```
//Crea el servicio de socket servidor con los datos
ServerSocket socketServidor = new ServerSocket(Puerto);
System.out.println("Esperando peticion de cliente...");
//En espera de petición de cliente
Socket socketCliente = socketServidor.accept();
```

Una vez realizada la conexión va a generar un número aleatorio del 0 al 100 y creará un flujo de entrada del socket para recibir los intentos del cliente para acertar el número, al recibirlo evalúa si es menor (le asignamos el 2), mayor (le asignamos el 1) o igual (le asignamos 0) para posteriormente escribirlo en el flujo de salida para que lo lea el cliente.

```
//Genera número aleatorio del 0 al 199 y lo muestra por pantalla
numeroAleatorio = (int) (Math.random() * 100);
System.out.println("Numero aleatorio: " + numeroAleatorio);
do {
    //Creamos un flujo de entrada del socket y leemos el entero de entrada
    DataInputStream flujoEntrada = new DataInputStream(socketCliente.getInputStream());
    numUsuario = flujoEntrada.readInt();
    //Comparamos números para saber si es mayor, menor o igual
    if (numUsuario < numeroAleatorio) {
        valor = 2;
    } else if (numUsuario > numeroAleatorio) {
        valor = 1;
    } else {
        valor = 0;
    }
    DataOutputStream flujoSalida = new DataOutputStream(socketCliente.getOutputStream());
    flujoSalida.writeShort(valor);
} while (valor > 0); //Repetimos bucle hasta que el valor sea 0 o correcto
```

Por último cerramos el socket y terminando así la conexión.

- Clase Cliente: Comenzamos conectando al servidor a través de socket con los datos del mismo.

```
public class Cliente {

    //Definición de constantes
    static final String Host = "localhost"; //Dirección local del servidor
    static final int Puerto = 2000; //Puerto del servidor

    /**
     * Constructor de la clase donde definimos que hará la clase cuando la
     * instanciamos
     */
    public Cliente() {
        int numUsuario;
        int numIntento = 1;
        short valor;

        System.out.println("\n¡ADIVINE EL NUMERO DEL 0 AL 100 GENERADO POR EL SERVIDOR!");
        System.out.println("-----");
        try {
            //Creamos el Socket con los datos del servidor
            Socket socketCliente = new Socket(Host, Puerto);
```

A continuación solicita al usuario que ingrese un número del 0 al 100 para luego mandárselo al servidor. Vamos a crear un método dentro de esta clase para solicitar el número y verificar que sea correcto:

```
public int peticionNumUsuario(int numIntento) {
    //Definición de variables y objetos
    Scanner teclado = new Scanner(System.in);
    int numUsuario = 0;
    boolean error;
    //Solicita el número hasta que sea correcto
    do {
        error = false;
        try {
            System.out.println("Intento " + numIntento + ": ");
            numUsuario = teclado.nextInt();
            if (numUsuario > 100) {
                throw new Exception();
            }
        } catch (Exception e) {
            System.err.println("El dato introducido no es correcto");
            teclado.next();
            error = true;
        }
    } while (error);

    return numUsuario;
}
```

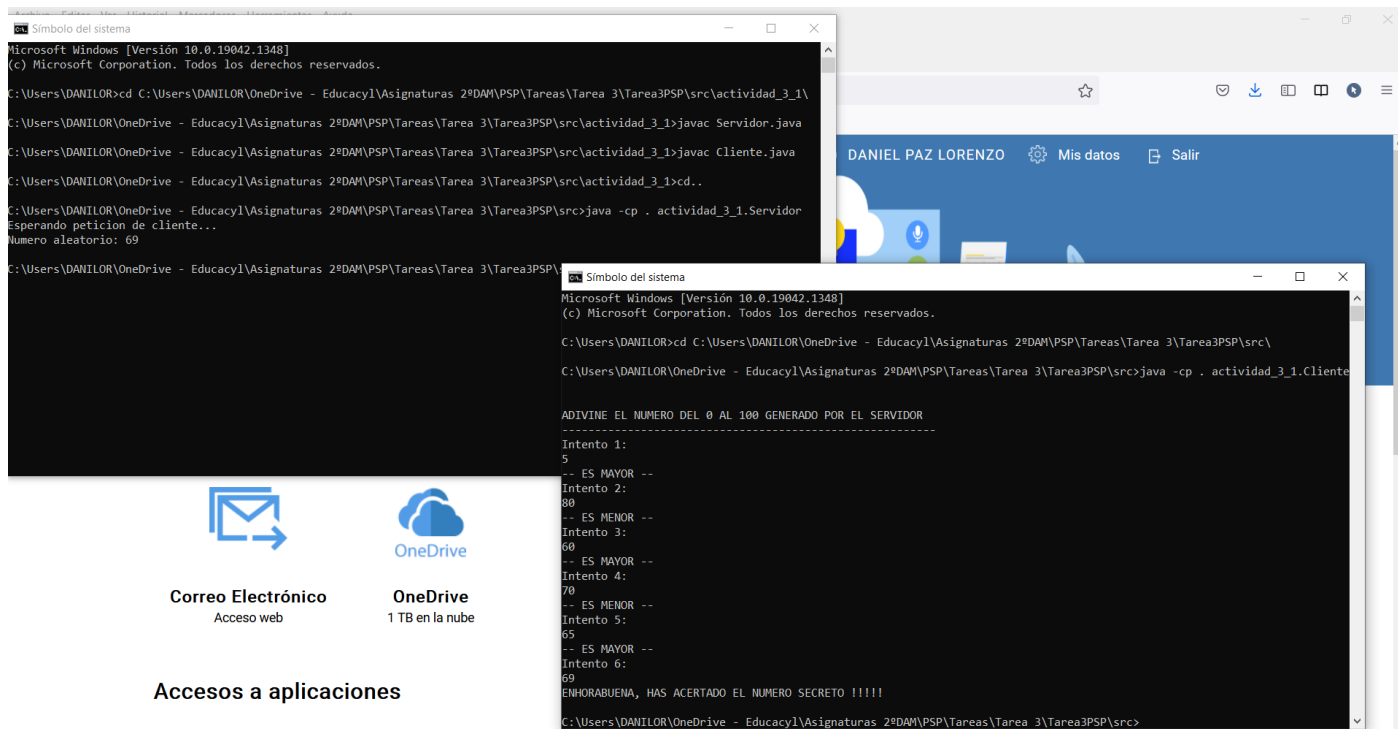
Una vez recogemos el número lo escribimos en un flujo de salida del socket para que el servidor lo lea, realice la comparación con el número que ha generado y devuelva, a través del flujo de entrada que hemos creado, la contestación de 0 (igual), 1 (menor) o 2 (mayor) que vamos a mostrar al usuario para que siga realizando intentos hasta que acierte el número, es decir hasta que “valor = 0”:

```
//Creamos un flujo de salida al socket y escribimos el número
//para que lo recoja el servidor
DataOutputStream flujoSalida = new DataOutputStream(socketCliente.getOutputStream());
flujoSalida.writeInt(numUsuario);
//Creamos un flujo de entrada del socket para recoger la respuesta del servidor
DataInputStream flujoEntrada = new DataInputStream(socketCliente.getInputStream());
valor = flujoEntrada.readShort();
//Evalúa la respuesta del servidor y muestra un mensaje por pantalla
if (valor == 2) {
    System.out.println("-- ES MAYOR --");
} else if (valor == 1) {
    System.out.println("-- ES MENOR --");
} else if (valor == 0) {
    System.out.println("ENHORABUENA, HAS ACERTADO EL NUMERO SECRETO !!!!!");
}
numIntento++;

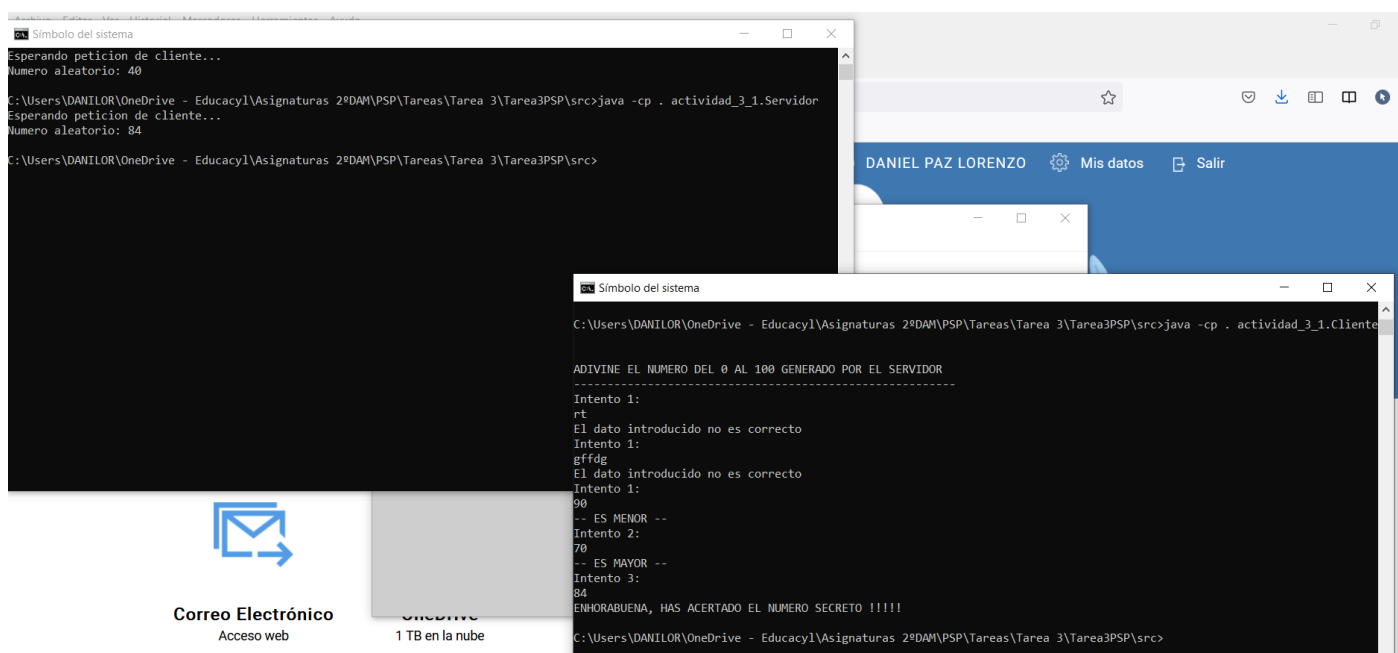
} while (valor > 0); //Repite bucle hasta que el valor sea correcto
//Cerramos el socket
socketCliente.close();
```

### Pruebas de aplicación de la actividad 3.1

Para realizar las pruebas lo primero que vamos a hacer será abrir dos terminales, en una vamos a ejecutar la clase servidor y en la otra la clase Cliente. En la primera terminal compilamos las 2 clases y ejecutamos la clase Servidor que se quedará a la espera de recibir la petición del cliente, mostrará por pantalla que está esperando al cliente y el número que ha generado de manera aleatoria. En la segunda terminal ejecutamos directamente la clase Cliente que nos va a ir solicitando números hasta que acertemos el que ha generado el Servidor:



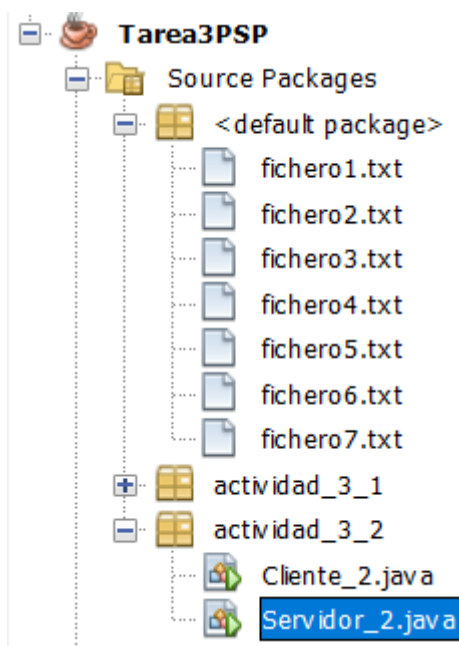
Otra prueba que haremos será la de introducir en la parte cliente datos que no sean válidos y podemos ver que la aplicación muestra que el dato introducido no es correcto y nos vuelve a solicitar otro sin incrementar el número de intentos:



### ACTIVIDAD 3.2

En este apartado se nos pide que creamos una aplicación del tipo cliente-servidor en la que el cliente realiza la petición de un fichero al servidor que comprueba si este existe dentro del proyecto y se lo manda al cliente. Si no existiera el fichero el servidor manda un mensaje informando del error al cliente.

Para la realización de esta actividad he optado también por crear 2 clases, una para la parte del servidor y la otra para el cliente, además he creado dentro del proyecto 7 ficheros con contenido del tipo \*.txt para poder probar el funcionamiento de la aplicación:



*Al nombre de las clases les he añadido el 2 para distinguirlas de la actividad anterior*

- Clase Servidor\_2: Dentro de esta clase vamos a crear el SocketServer con el puerto 1500, como nos pide la tarea y a continuación un socket que se quedará a la espera de la petición del cliente antes de conectarse:

```
public class Servidor_2 {

    ///Declaración de constantes con el puerto de conexión del servicio
    static final int Puerto = 1500;

    public Servidor_2() {
        //Declaramos variables y objetos de clase
        File ficheroServidor;
        String nombreUsuarioFichero, mensajeServidor;
        int in;

        try {
            //Creamos socket del servidor con el puerto 1500
            ServerSocket socketServidor = new ServerSocket(Puerto);
            //Espera la petición de cliente y cuando entra la acepta
            System.out.println("Esperando petición de cliente...");
            Socket socketCliente = socketServidor.accept();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Una vez aceptada la conexión continuaría recibiendo el nombre del fichero que desea el cliente a través de la lectura del flujo de entrada socket que hemos creado. Comprueba si el fichero existe dentro del proyecto, si es así lo lee y se lo manda al cliente a través del flujo de salida socket creado y si por el contrario no existiera le manda un mensaje al cliente, también a través del flujo salida socket, de que el fichero no existe dentro del proyecto:

```
//Crea un flujo de entrada del socket para recoger el fichero enviado por el cliente
DataInputStream flujoEntrada = new DataInputStream(socketCliente.getInputStream());
nombreUsuarioFichero = flujoEntrada.readUTF();
//Creamos un flujo de salida del socket
DataOutputStream flujoSalida = new DataOutputStream(socketCliente.getOutputStream());
ficheroServidor = new File(nombreUsuarioFichero);
System.out.println("Fichero solicitado por el cliente: " + ficheroServidor);
if (ficheroServidor.exists()) { //Comprueba si existe el fichero
    //Manda mensaje de que existe el fichero
    mensajeServidor = "El fichero existe y contiene lo siguiente: \n\n";
    System.out.println(mensajeServidor);
    flujoSalida.writeUTF(mensajeServidor);
    //Lee el fichero y lo escribe en el flujo de salida además de imprimir por pantalla
    BufferedInputStream bis = new BufferedInputStream(new FileInputStream(ficheroServidor));
    while ((in = bis.read()) != -1) {
        flujoSalida.write(in);
        System.out.print((char) in);
    }
} else {
    //Si no existe fichero también manda mensaje informando
    mensajeServidor = "El fichero no existe";
    System.out.println(mensajeServidor);
    flujoSalida.writeUTF(mensajeServidor);
}
```

*Una vez finalizada la aplicación se cierra la conexión*

- Clase Cliente\_2: En la parte cliente creamos el socket con los datos del servidor para conectarnos a él, a continuación solicitamos al usuario que ingrese el nombre del fichero que desea consultar, lo recogemos en un String y se lo enviamos al servidor a través del flujo de salida socket.

En ese momento se quedará leyendo por el flujo de entrada socket a que el servidor le devuelva respuesta de si existe o no el fichero y en caso afirmativo muestra por pantalla toda el contenido del fichero que nos manda el servidor haciendo uso de un bucle while. Una vez hemos terminado de mostrar el fichero se cierra la conexión y por lo tanto también la aplicación:

```
public Cliente_2() {
    //Declaración de variables y objetos
    String nombreFichero;
    Scanner teclado = new Scanner(System.in);
    int caracterAsciiEntrada;

    try {
        //Creamos la conexión al servidor
        Socket socketCliente = new Socket(Host, Puerto);
        //Solicita el nombre del fichero y lo recoge en un String
        System.out.println("Introduzca el nombre del fichero --> ");
        nombreFichero = teclado.nextLine();
        //Creamos un flujo de salida del Socket y escribimos en él el nombre del fichero
        DataOutputStream flujoSalida = new DataOutputStream(socketCliente.getOutputStream());
        flujoSalida.writeUTF(nombreFichero);
        //Creamos un flujo de entrada del socket y mostramos por pantalla la información leída
        DataInputStream flujoEntrada = new DataInputStream(socketCliente.getInputStream());
        System.out.println(flujoEntrada.readUTF());
        //Recoge cada byte del flujo de entrada y lo muestra por pantalla
        while ((caracterAsciiEntrada = flujoEntrada.read()) != -1) {
            System.out.print((char) caracterAsciiEntrada);
        }

        socketCliente.close();
    }
```

### Pruebas de aplicación de la actividad 3.2

Para las pruebas de la aplicación al ser del tipo cliente-servidor tendremos que ejecutar primero la clase Servidor\_2 en una terminal y a continuación la clase Cliente\_2 en otra terminal. Una vez la hayamos compilado y ejecutado por separado podemos ver que se conectan y en la parte cliente nos pedirá que demos el nombre del fichero que queremos que nos muestre. Dentro del proyecto hay 7 ficheros que he creado anteriormente con los nombres de “fichero1...7.txt” voy a pedirle a la aplicación que me muestre varios y por último probaré también que es lo que pasa si le doy un nombre de fichero que no existe dentro del proyecto:

```

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src\actividad_3_2>javac Servidor_2.java
C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src\actividad_3_2>javac Cliente_2.java
C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src\actividad_3_2>cd..
C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>java -cp . actividad_3_2.Servidor_2
Esperando petición de cliente...
Fichero solicitado por el cliente: fichero1.txt
El fichero existe y contiene lo siguiente:

Hola Cliente, este es el contenido del fichero 1 del servidor,
contenido dentro de la carpeta ficheroServidor.

Despues de mostrar el contenido de este fichero finalizara la conexion con el servidor.
Gracias y un saludo.

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>
  
```

```

C:\Users\DANILOR>cd C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src
C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>java -cp . actividad_3_2.Cliente_2
Introduzca el nombre del fichero -->
fichero1.txt
El fichero existe y contiene lo siguiente:

Hola Cliente, este es el contenido del fichero 1 del servidor,
contenido dentro de la carpeta ficheroServidor.

Despues de mostrar el contenido de este fichero finalizara la conexion con el servidor.
Gracias y un saludo.

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>
  
```

*A la izquierda ejecutamos el Servidor\_2 y a la derecha Cliente\_2, le hemos solicitado el contenido de fichero1.txt y es el que se muestra por pantalla*

```

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>java -cp . actividad_3_2.Servidor_2
Esperando petición de cliente...
Fichero solicitado por el cliente: fichero5.txt
El fichero existe y contiene lo siguiente:

Hola Cliente, este es el contenido del fichero 5 del servidor,
contenido dentro de la carpeta ficheroServidor.

Despues de mostrar el contenido de este fichero finalizara la conexion con el servidor.
Gracias y un saludo.

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>
  
```

```

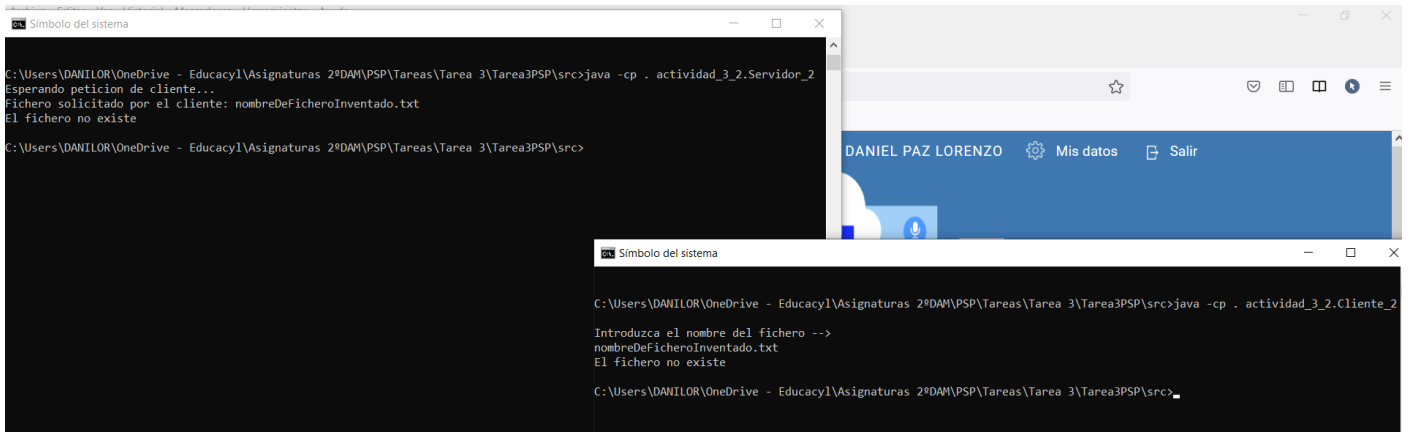
C:\Users\DANILOR>cd C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src
C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>java -cp . actividad_3_2.Cliente_2
Introduzca el nombre del fichero -->
fichero5.txt
El fichero existe y contiene lo siguiente:

Hola Cliente, este es el contenido del fichero 5 del servidor,
contenido dentro de la carpeta ficheroServidor.

Despues de mostrar el contenido de este fichero finalizara la conexion con el servidor.
Gracias y un saludo.

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>
  
```

*Hacemos lo mismo pero ahora le vamos a solicitar el fichero5.txt y vemos que el resultado es distinto porque ha cambiado el contenido, ahora nos indica que es el fichero 5*



The screenshot shows a web browser window displaying a user profile for 'DANIEL PAZ LORENZO' with options for 'Mis datos' and 'Salir'. In the background, two Windows command prompt windows are open. The top window shows the execution of a Java server program that is waiting for a client and then reports that a requested file does not exist. The bottom window shows the execution of a Java client program that prompts for a file name and also reports that the file does not exist.

```
C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>java -cp . actividad_3_2.Servidor_2
Esperando petición de cliente...
Fichero solicitado por el cliente: nombreDeFicheroInventado.txt
El fichero no existe

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>java -cp . actividad_3_2.Cliente_2
Introduzca el nombre del fichero -->
nombreDeFicheroInventado.txt
El fichero no existe

C:\Users\DANILOR\OneDrive - Educacyl\Asignaturas 2ºDAM\PSP\Tareas\Tarea 3\Tarea3PSP\src>
```

*Como última prueba mandamos un nombre de fichero que no existe dentro del proyecto y vemos que nos devuelve un mensaje diciendo que efectivamente ese fichero no existe*