

YOURCLOUDBOX

1º Descripción general del proyecto

Simular un FTP, un sistema de subida y bajada de ficheros para usuarios registrados. Sería como una nube donde cada usuario registrado tiene un espacio de almacenamiento asignado para descargar y subir ficheros. Sería un Dropbox con funciones limitadas.

Constaría de 5 operaciones básicas:

- Registro: registro de nuevos usuarios asignándole un pequeño espacio en la nube dedicado para el mismo.
- Login: inicio de sesión del usuario previamente registrado.
- Upload: subida de los archivos en el espacio de la nube asignado al usuario registrado.
- Download: bajada de los archivos alojados en el espacio de la nube dedicado al usuario.
- Listar: lista los ficheros que ya están subidos en el espacio asignado al usuario

La aplicación estará basada en una arquitectura cliente-servidor y utilizará sockets para la comunicación entre ambos.

Una vez que un cliente inicia una comunicación tendrá un protocolo para identificar cada una de las operaciones.

Protocolos de comunicación cliente-servidor:

OPERACIÓN	CLIENTE
Registro	#register#user#password
Login	#login#user#password
Upload	#upload#user#namefile#totalsize #upload-content#user#bytes#numberbytes
Download	#download#user#namefile #download-content#user#numberbytes
Listar	#listar#user
Eliminar	#delete#user#filename

OPERACIÓN	SERVIDOR
Registro	#register#useralreadyexist #register#usercreated
Login	#login#ok #login#fail
Upload	#upload#namefile #upload-content#numberbytes #upload#error#mensaje
Download	#download#namefile#totalsize #download-content#bytes#numberbytes #download#error#mensaje
Listar	#listar#filename1, filename2...
Eliminar	#delete#mensaje

2º Descripción de los módulos/unidades funcionales que componen el proyecto.

El producto se compondrá de un servidor (sistema encargado de almacenar los ficheros de cada uno de los clientes) y de un cliente (sistema encargado de iniciar la comunicación con el servidor permitiendo el intercambio de ficheros).

3º Diagrama UML o similar describiendo de qué forma se relacionan los diferentes módulos.

Es importante aclarar que esto se trata de un primer diseño del sistema. Se aportará el diagrama completo al finalizar el proyecto, por tanto, está sujeto a cambios. El sistema se compone de dos aplicaciones: cliente y servidor.

Clases de la aplicación cliente:

- comunicacionManagerCliente: se encarga de crear el socket para la comunicación con el servidor.

- protocoloManager: interpretar y crear los mensajes que se enviarán al servidor
- fileManager: manejo de ficheros

Clases de la aplicación servidor:

- comunicacionManagerServidor: se encarga de recibir y procesar las peticiones del cliente.
- protocoloManager: interpretar y crear los mensajes que se enviarán al cliente
- fileManager: manejo de ficheros

4º Lenguajes de programación, plataformas de desarrollo, sistemas operativos, etc. que vais a utilizar en el desarrollo.

Se desarrollará con lenguaje de programación Java 17, usando el IDE Eclipse versión 2022-09, bajo el sistema operativo Windows 10.

5º Planificación

En este apartado se va a subdividir el desarrollo del software en diferentes historias de usuario. Y estas, a su vez, en tareas. Una historia de usuario se considerará lista para entregar cuando todas sus tareas estén finalizadas.

Historia de usuario 1 (4 horas): Registro de usuarios

- Como responsable de la empresa YOURCLOUBOX quiero que los usuarios puedan darse de alta en la aplicación con el objetivo de que accedan a su uso seguro.
 - 1) Implementar un cliente que permita la comunicación con un servidor mediante socket.

- 2) Implementar un servidor que atienda las peticiones de los clientes mediante socket.
- 3) Implementación del registro: el cliente envía la cadena (#register#user#password) y una vez en el servidor se buscaría ese usuario en un fichero. Si existe el usuario el servidor respondería (#register#useralreadyexist) y si no está (#register#usercreated) para que usuario y password queden grabados en el fichero.
- 4) Pruebas integradas.

Historia de usuario 2 (3 horas): Login de usuarios

- Como responsable de la empresa YOURCLOUBOX quiero que los usuarios puedan acceder a la aplicación mediante identificación para su uso seguro.
- 1) Implementar el login basado en el usuario y password. El cliente envía el mensaje (#login#user#password) y el servidor responde si es usuario estaba previamente registrado y credenciales son correctas (#login#ok) o responde (#login#fail) si no coinciden las credenciales.
 - 2) Pruebas integradas

Historia de usuario 3 (16 horas): Upload

- Como responsable de la empresa YOURCLOUBOX quiero que los usuarios puedan subir ficheros a su espacio asignado para almacenar sus ficheros en la nube.
- 1) Implementar la subida de ficheros. El cliente envía el fichero (#upload#user#namefile) dividido en un conjunto de bytes y el servidor recompone el fichero.
 - 2) Pruebas integradas.

Historia de usuario 4 (16 horas): Download

- Como responsable de la empresa YOURCLOUBOX quiero que los usuarios puedan descargar ficheros desde su espacio asignado para recuperar sus ficheros.

- 1) Implementar la descarga de ficheros. El servidor envía el fichero dividido en un conjunto de bytes y el cliente recompone el fichero (#download#user#namefile).
- 2) Pruebas integradas.

Historia de usuario 5 (4 horas): Listar

- Como responsable de la empresa YOURCLOUBOX quiero que los usuarios puedan visualizar qué ficheros tienen en su espacio asignado para que puedan conocer su información almacenada.
- 1) Implementar la visualización de ficheros. El servidor envía al cliente información del conjunto de ficheros (#listar#user) que tiene almacenado el usuario en su espacio asignado.
 - 2) Pruebas integradas.

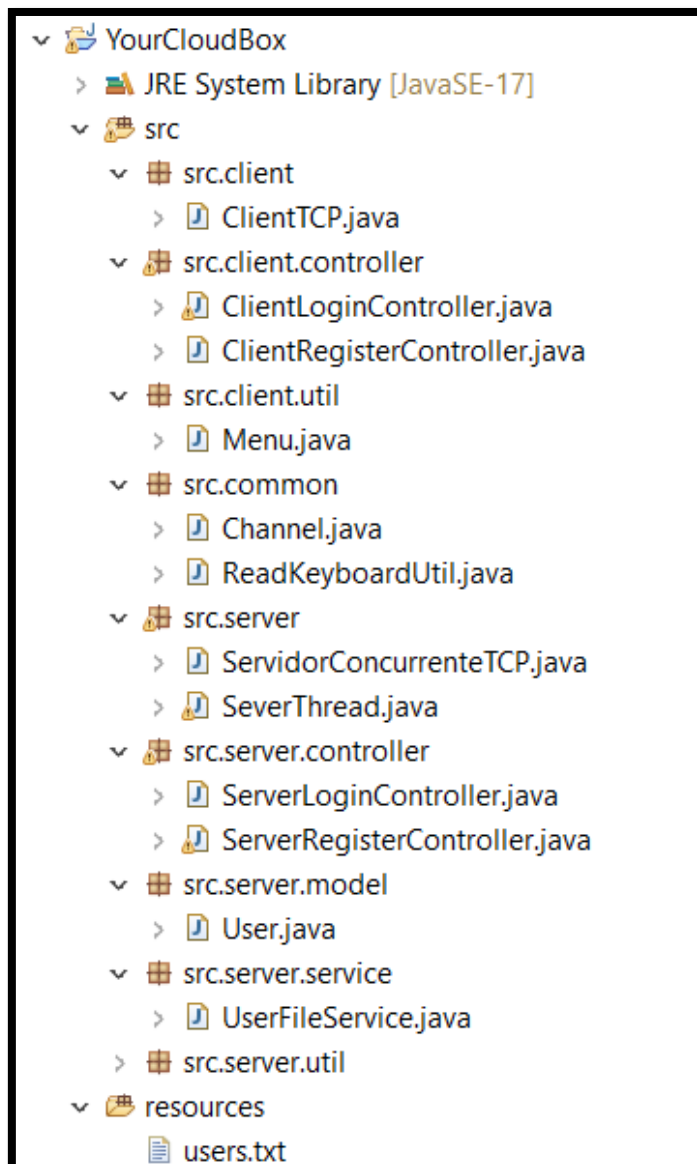
Historia de usuario 6 (4 horas): Eliminar

- Como responsable de la empresa YOURCLOUBOX quiero que los usuarios puedan eliminar sus propios ficheros para que puedan liberar espacio.
- 1) Implementar la eliminación de ficheros. El cliente envía al servidor el nombre del fichero listado a eliminar (#delete#user#filename).
 - 2) Pruebas integradas.

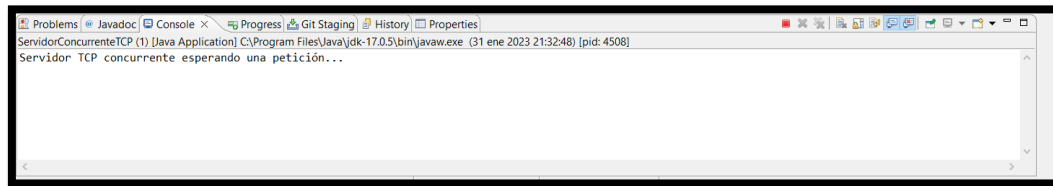
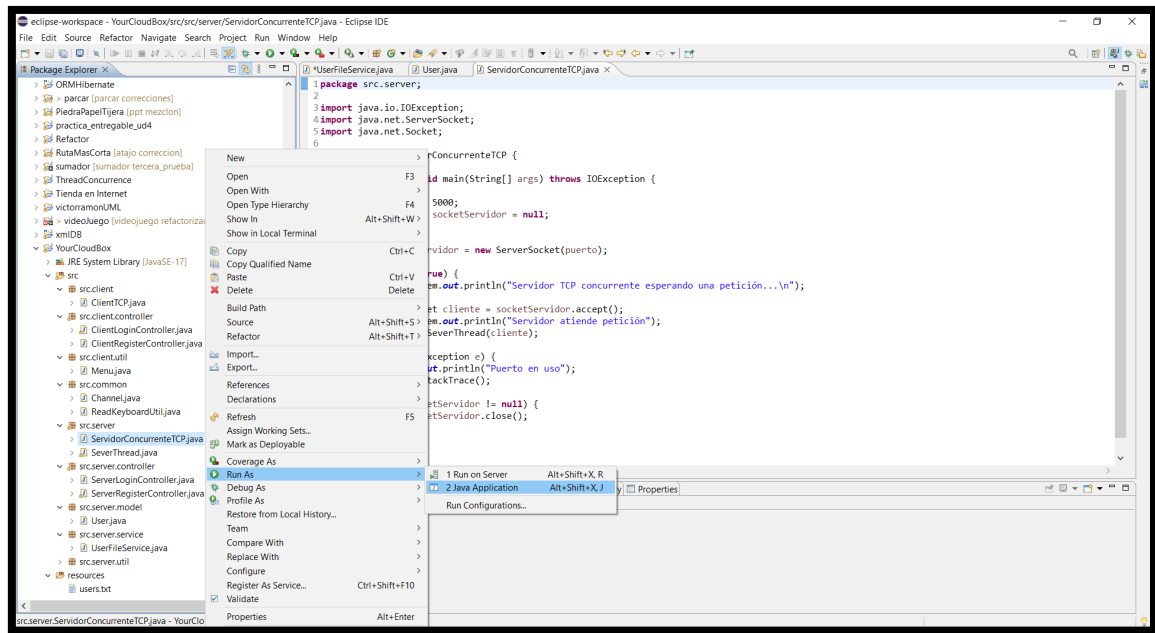
6ª Primera entrega de lo desarrollado

He desarrollado la estructura inicial del proyecto y el registro de usuarios que ya está operativo. Otras partes están iniciadas, pero no están operativas. De momento, si se quiere probar el proyecto sería el registro de usuarios.

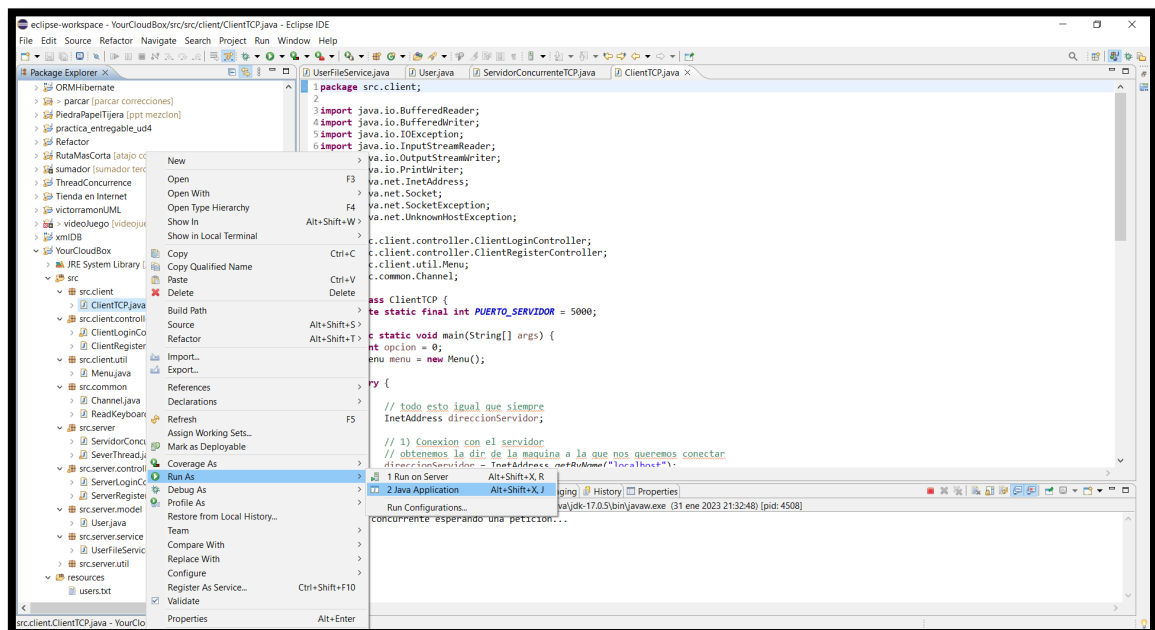
En el archivo “users.txt” se almacenarán los usuarios y contraseñas de la aplicación.



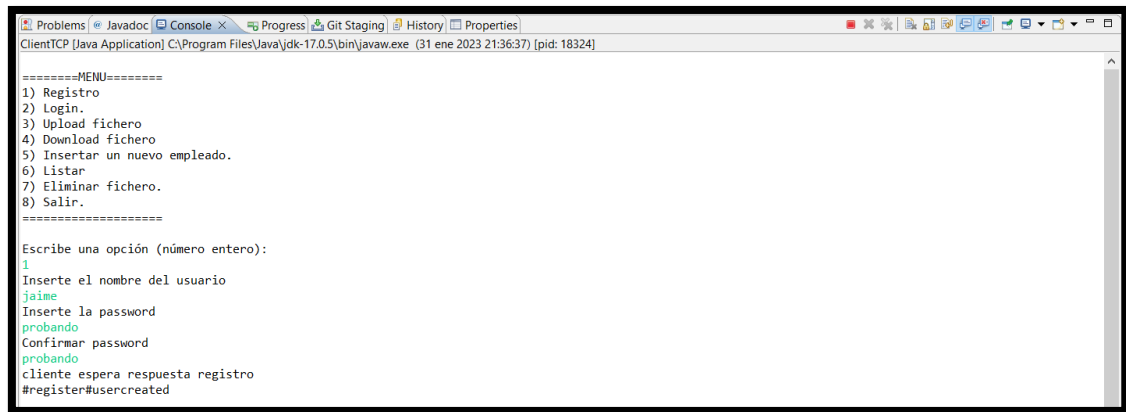
Primero hay que ejecutar la clase `ServidorConcurrenteTCP` que representa al servidor y si se ejecuta correctamente se imprimirá por pantalla que está esperando la petición.



Segundo hay que ejecutar la clase `ClientTCP` que llamará al menú de opciones.



Del menú de opciones solo está implementado el registro, del cual muestro un ejemplo de su funcionamiento creando al usuario “Jaime” y su password “probando”. Como dicho usuario no existe, queda registrado.

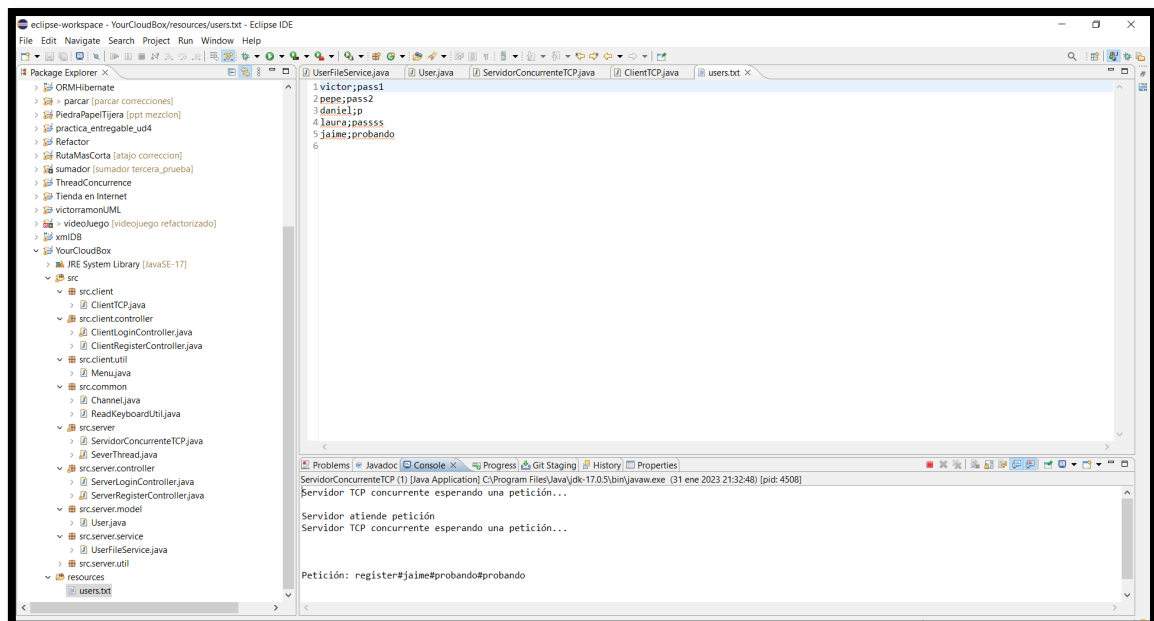


```

=====MENU=====
1) Registro
2) Login.
3) Upload fichero
4) Download fichero
5) Insertar un nuevo empleado.
6) Listar
7) Eliminar fichero.
8) Salir.
=====

Escribe una opción (número entero):
1
Inserte el nombre del usuario
jaime
Inserte la password
probando
Confirmar password
probando
cliente espera respuesta registro
#register#usercreated
  
```

Podemos comprobar en el archivo “users.txt” que se ha registrado correctamente y que en la consola del servidor se ha recibido la petición.



```

1 victor;pass1
2 pepe;pass2
3 daniel;p
4 laura;passsss
5 jaime;probando
6
  
```

```

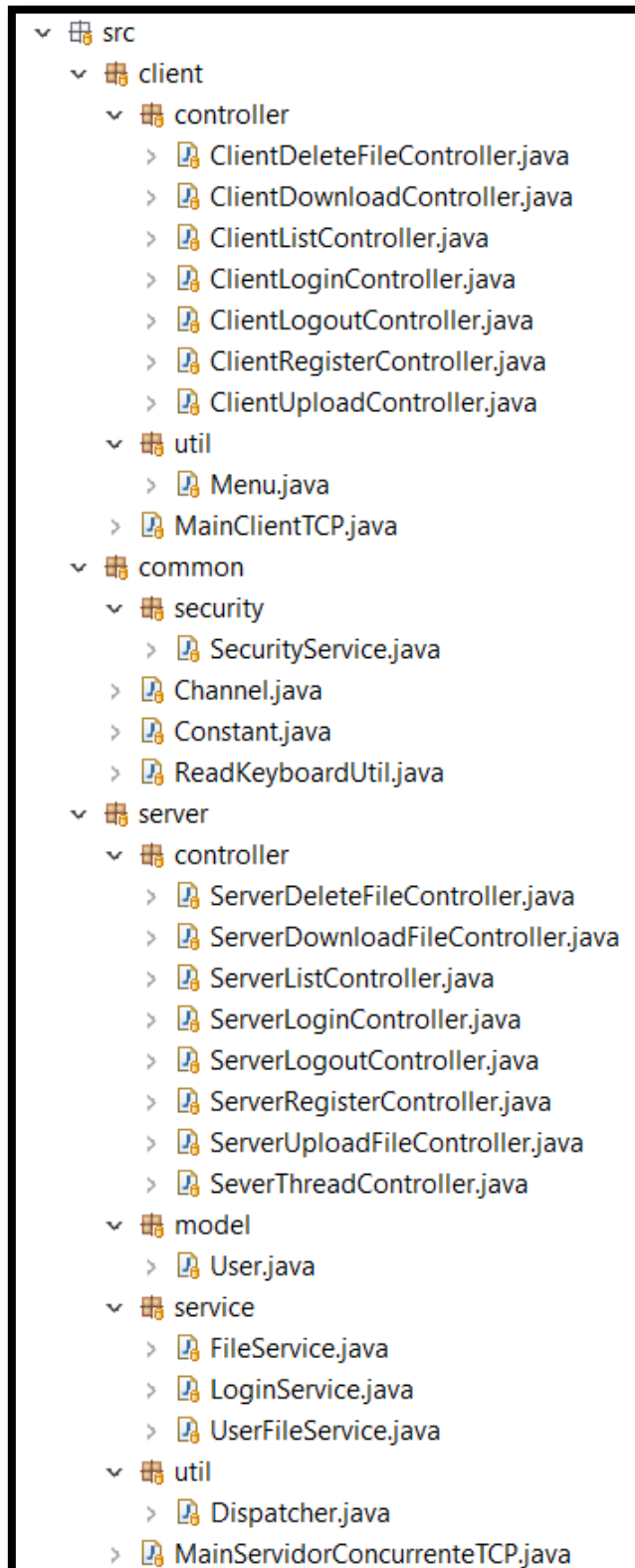
ServidorConcurrenteTCP (1) [Java Application] C:\Program Files\Java\jdk-17.0.5\bin\javaw.exe (31 ene 2023 21:32:48) [pid: 4508]
Servidor TCP concurrente esperando una petición...

Servidor atiende petición
Servidor TCP concurrente esperando una petición...

Petición: register#jaime#probando#probando
  
```


7º ENTREGA FINAL - Manual de uso

He completado la estructura final del proyecto con 7 opciones de menú operativas.



```

=====MENU=====
1) Registro
2) Login.
3) Upload fichero
4) Download fichero
5) Listar
6) Eliminar fichero.
7) Salir.
=====

```

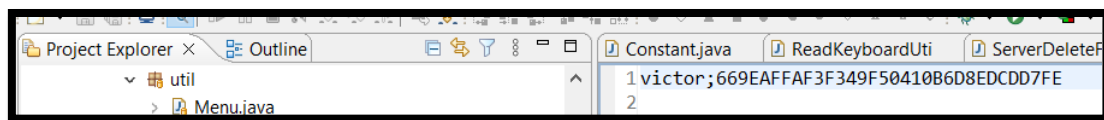
La estructura se divide en 3 paquetes principales subdivididos en otros:

1º) Client

- controller: aloja las clases de interacción (eventos) que hace el usuario.
- util: utilidad que contine el menú con el que interactúa el cliente.
- Main (clase): arranca el cliente

2º) Common

- security: algoritmo de codificación M5D, para que la base datos de usuarios registrados tenga las contraseñas cifradas.

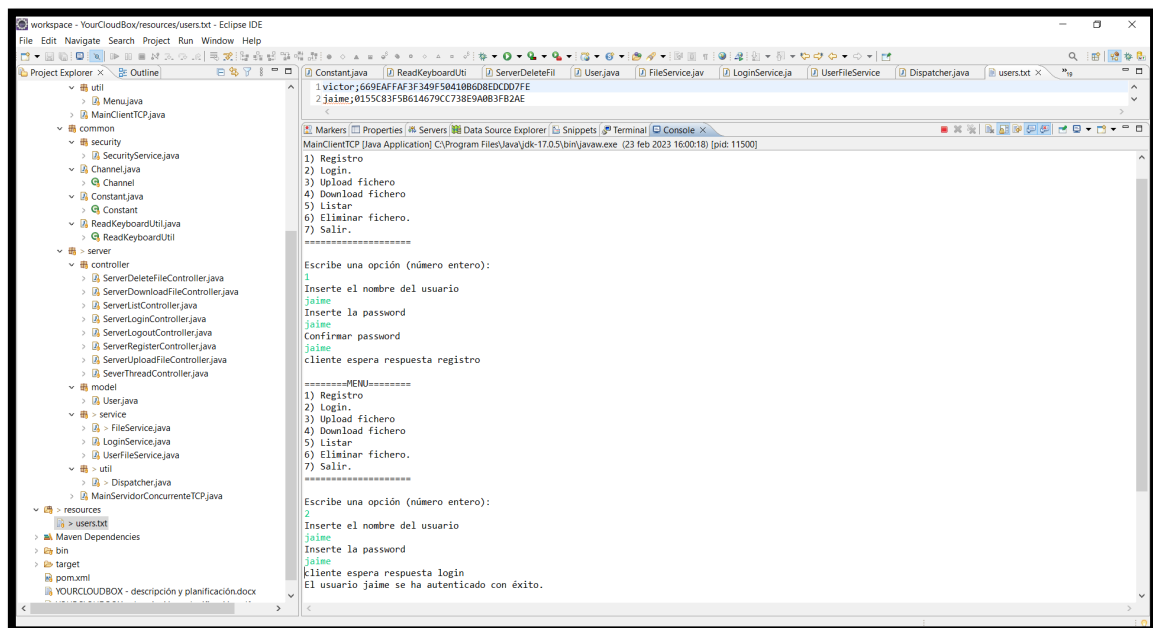


- Channel: encapsula los elementos usados en la comunicación entre el cliente y el servidor.
- Constant: directorios del servidor y del cliente.
- ReadKeyboardUtil: utilidad para entrada de datos por teclado.

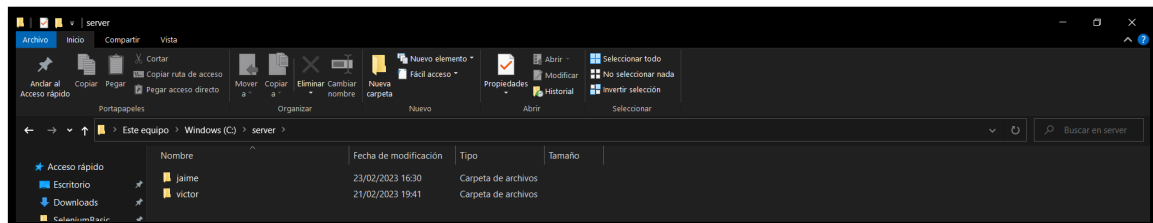
3º) Server

- controller: aloja las clases de interacción (eventos) que hace el servidor.
- model: constructor de usuarios.
- service: servicios de gestión de ficheros, usuarios y login.
- util: canaliza la petición del cliente al controlador del evento correspondiente.
- Main (clase): arranca el servidor

A continuación se muestra la funcionalidad de registro y login de usuario.



En el directorio se crea una nueva carpeta con el nombre del nuevo usuario como su espacio de almacenamiento.



A continuación se muestra cómo subir y bajar un fichero.

