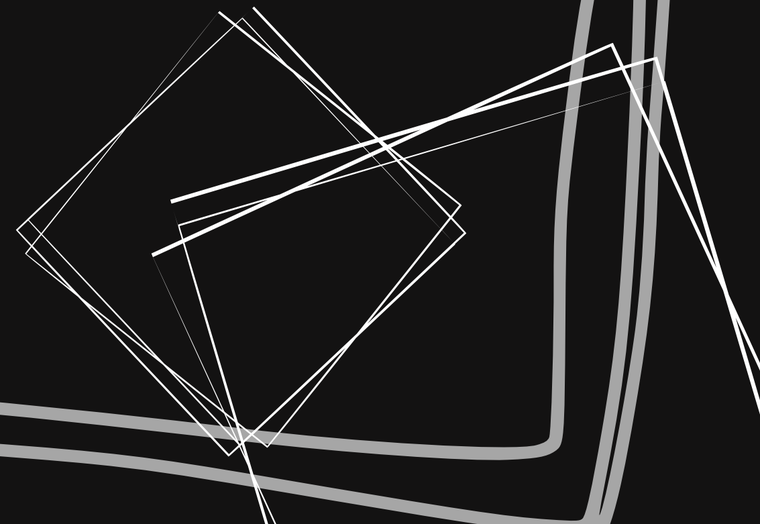
PROGETTO SOFTWARE BASATO SUI SOCKET

Realizzazione di un file server

made by :
Lucon , Borsato , Luo, Fiorini

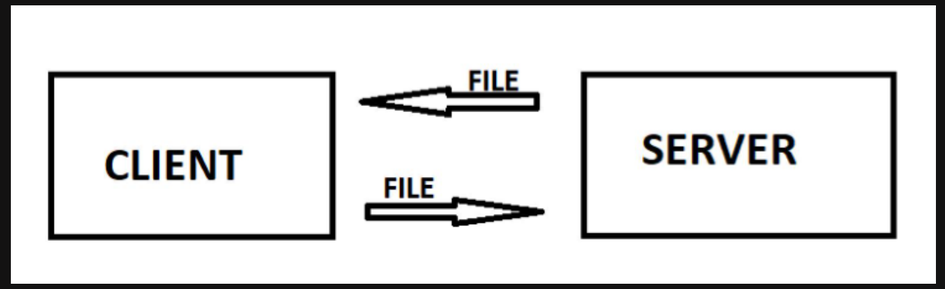


OBIETTIVO DEL PROGRAMMA:

Creare una applicazione C# che permetta di navigare il filesystem locale per poter effettuare l'upload di file su un server

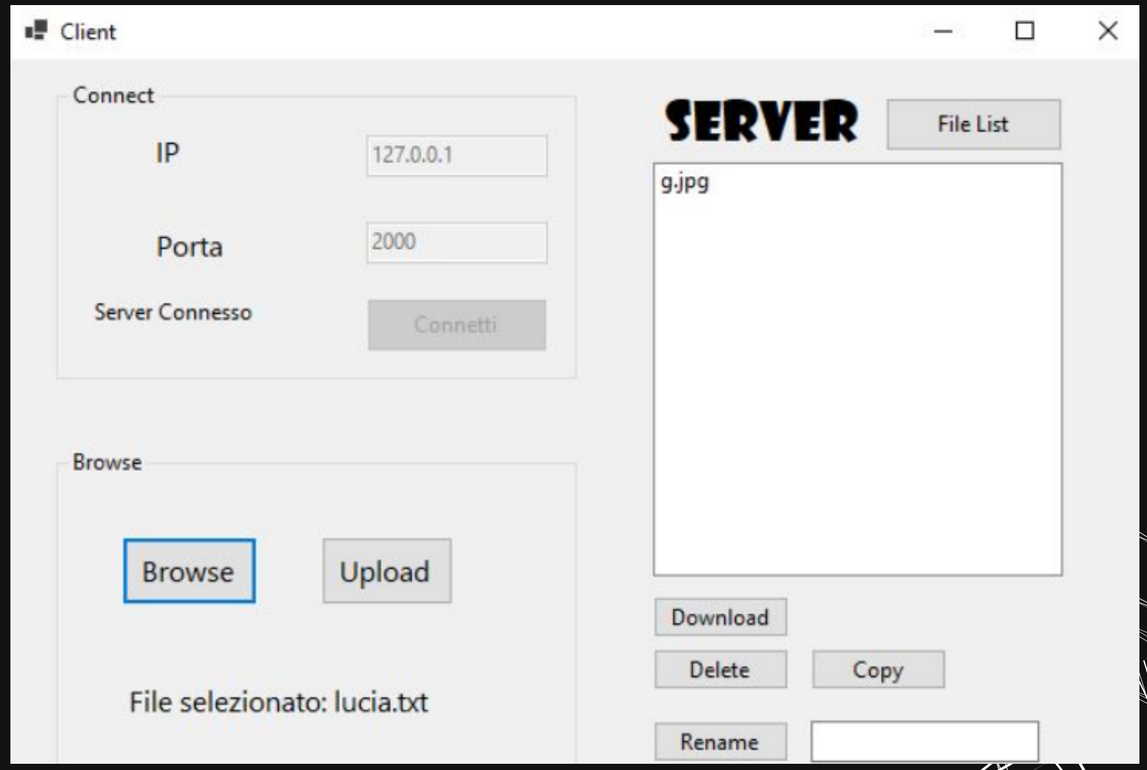
OBIETTIVO SECONDARIO:

Migliorare l'applicazione in modo che somigli un client FTP al fine di navigare sia il filesystem locale che quello remoto



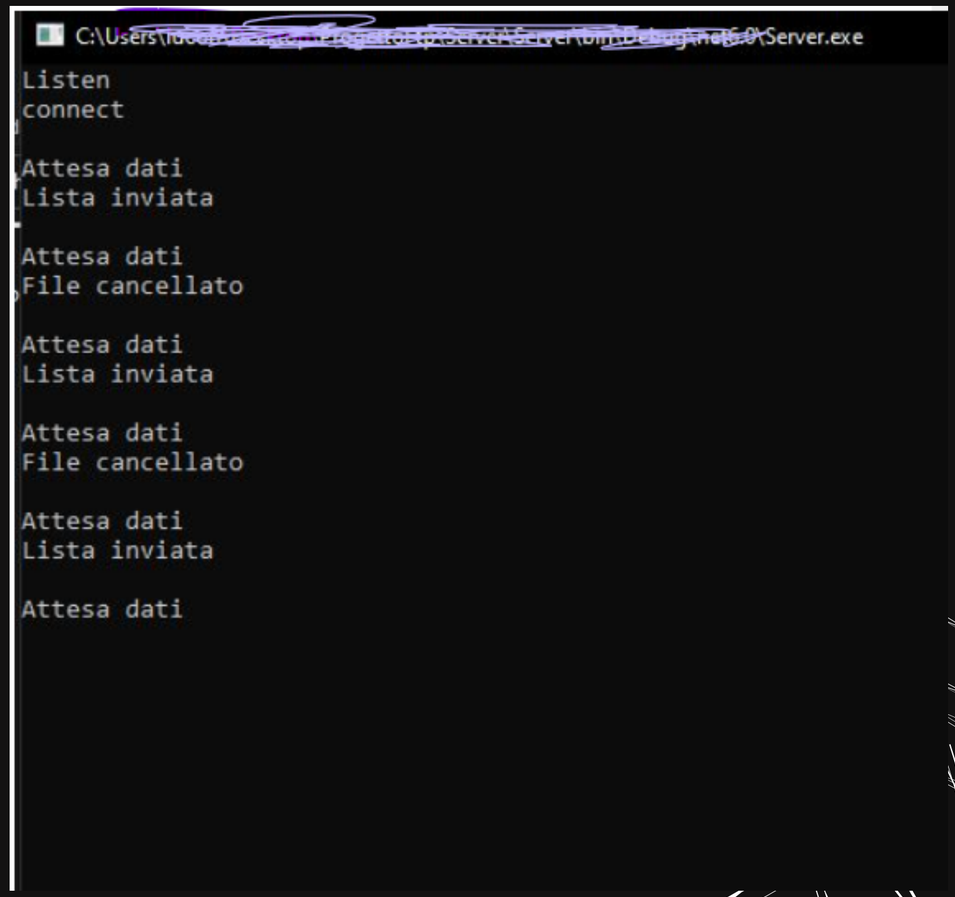
Funzionalità Implementate:

Il Client permette di connettersi a un server qualsiasi fornendo IP e porta, dopo aver connesso il client con il server sarà possibile effettuare l'upload di un file sul fileSystem del server.



I file possono avere una dimensione massima di 5MB.

Organizzazione Del Codice:



Client:

Il client possiede un form e una classe Connection, contiene i metodi collegati al form che permettono di trasferire file e informazioni con il server.

Server:

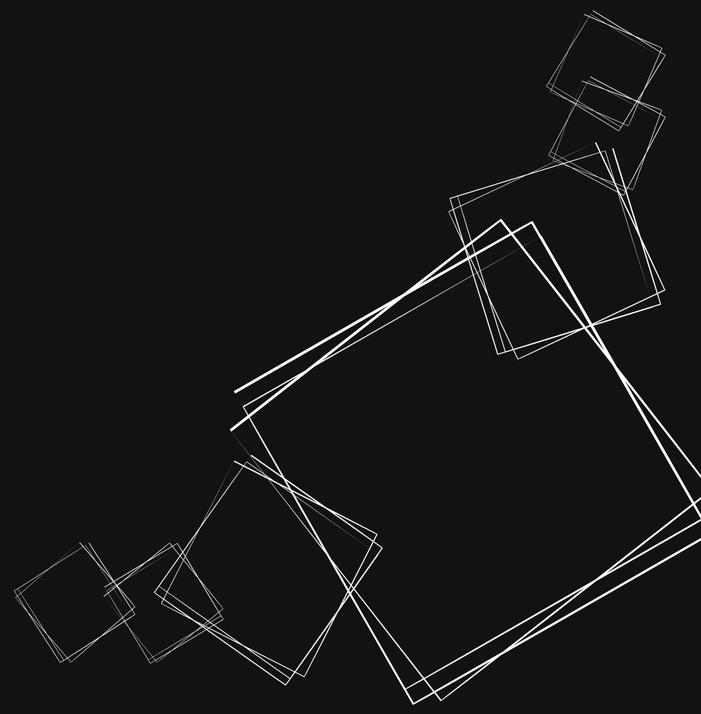
Il server è un'applicazione Console la classe Connection, contenente i metodi che permettono di gestire le varie richieste del client

Problematiche Rilevate:

Come poter gestire le richieste del client per il server?

La soluzione che abbiamo trovato a questo problema è di inserire, per ogni richiesta, come primo byte una lettera identificativa della richiesta, seguendo questo schema:

File	f	Il Client ha inviato un file al Server
Delete	d	Il Client vuole eliminare un file del Server
Request	r	Il Client ha richiesto l'invio di un file
Copy	c	Il Client vuole fare una copia di un file del Server
List	l	Il Client richiede la lista dei file del Server



Come trasferire un file?

Abbiamo creato un protocollo univoco per client e server.
per l'upload di un file i byte che trasmetterà il client saranno sempre composti da:

Primo Byte : **Lettera identificativa (f)**

Da Secondo a Quinto Byte: **Lunghezza del nome del file (N)**

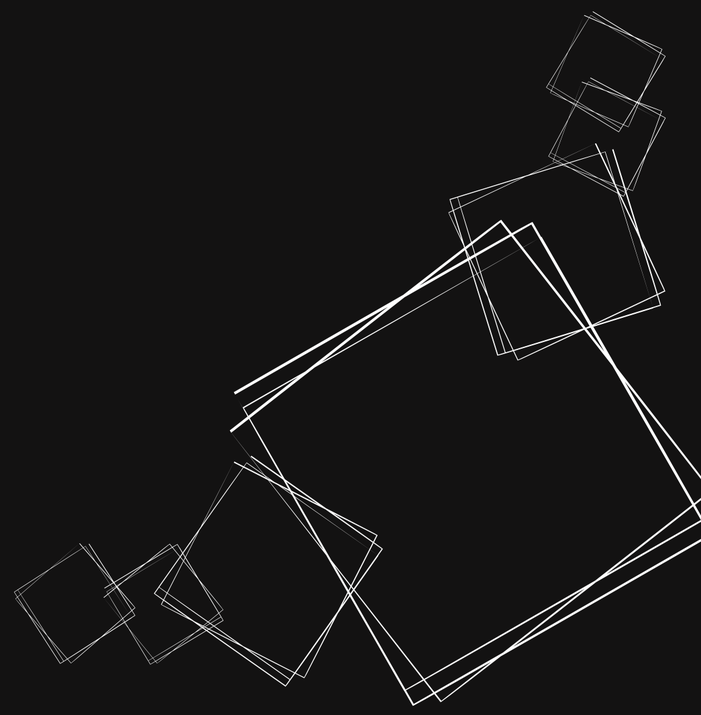
Da Sesto a N Byte: **Nome del file**

Da N+1 Byte: **Contenuto del file**

Per il download il server comporrà l'invio come visto precedentemente,
e il client manderà un messaggio contenente:

Primo Byte: **Lettera identificativa (r)**

Da Secondo a Ultimo Byte: **Nome file**



Eliminazione E Copia File

Quando verrà richiesta un’eliminazione di un file del server i byte saranno composti così:

Primo Byte: **Lettera identificativa (d/c)**

Da Secondo a Ultimo Byte: **Nome file**

Richiesta Lista Dei File

Quando il client chiederà al server di trasmettergli la lista dei file la richiesta sarà composta così:

Primo Byte: **Lettera identificativa (l)**

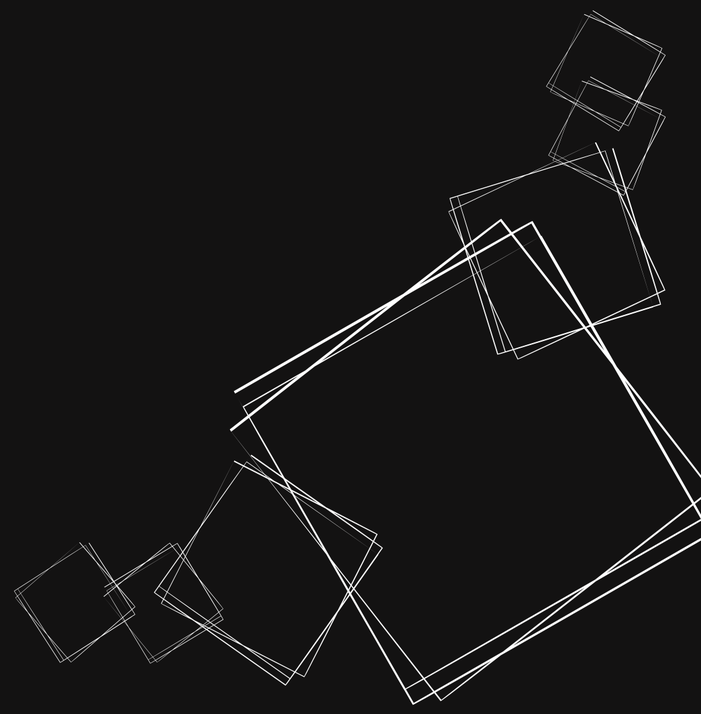
Il Server risponderà in questo modo:

Da Secondo a Quinto: **Lunghezza nome del primo file (N)**

Dal Sesto a N : **Nome primo file**

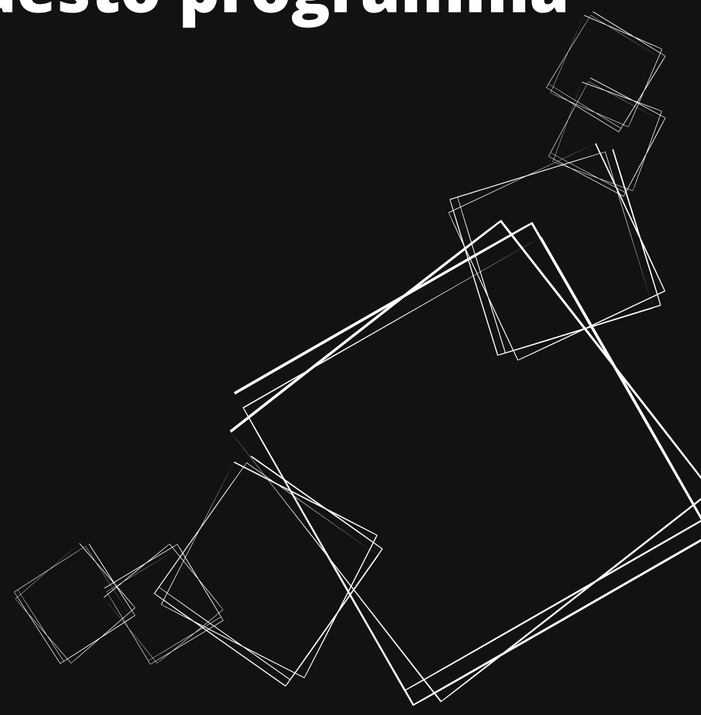
Da N+1 a N+4: **Lunghezza nome del secondo file (V)**

Da N+5 a V: **Nome secondo file:**
...
Continuando secondo questo schema il server invierà tutti i file disponibili



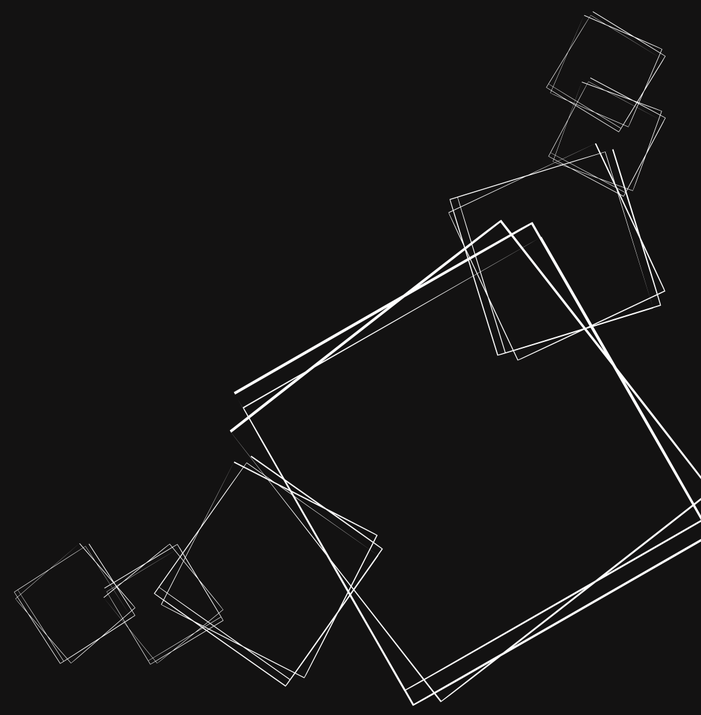
Sviluppi Futuri:

Al momento il Server può gestire solamente file di qualunque tipo, è possibile però in futuro adattare questa applicazione alla gestione anche di cartelle e sottocartelle, rendendo quindi questo programma sempre più simili a un Server FTP.



Fonti Utilizzate:

c# stream file multi threaded socket tcp ip send and receive file client server



GRAZIE PER AVERCI PRESTATO ATTENZIONE