



Istituto Tecnico Tecnologico Statale "Alessandro Volta"

Scuol@2.0

Chimica, Materiali e Biotecnologie
Grafica e Comunicazione
Trasporti e Logistica

Via Assisana, 40/E - loc. Piscille - 06135 Perugia
Centralino 075.31045 fax 075.31046 C.F. 80005450541
www.avolta.pg.it
voltauffici@tin.it dirigente@avolta.pg.it

Meccanica, Meccatronica ed Energia
Elettronica ed Elettrotecnica
Informatica e Telecomunicazioni

Corso di
Tecnologie e progettazione di sistemi informatici e
di telecomunicazioni

Classi 5AInf, 5BInf, 5CInf
a.s. 2016/2017

Relazione progetto
"TCPChat"

Studente: Nicolò Vescera

Docente/i: Monica Ciuchetti, Luca Mencagli

Documento di Specifica dei Requisiti (Software Requirements Specification)

1. Introduzione e formulazione del problema

Sviluppare un'applicazione client/server Java in grado di simulare una chat multi utente. Ogni utente deve potersi connettere ad un server centrale che gestisce le connessioni, deve poter mandare e ricevere messaggi da tutti gli altri utenti connessi, inviare e ricevere file e inviare e ricevere smile.

2. Descrizione dell'architettura dell'applicazione (componenti hardware e software)

Questa applicazione è stata sviluppata e testata utilizzando la JDK 1.8.0_121 fornita da Oracle e l'IDE Netbeans 8.2

2.1 Attori

In questa applicazione sono presenti 2 attori:

- Il **server**, si occupa di gestire la connessione dei vari client, di inoltrare i messaggi e i file che gli arrivano a tutti gli altri client, memorizza i vari nick dei client e assegnare un colore ad ogni nick
- Il **client** può connettersi al server, scegliere il proprio nick che gli sarà associato per tutta la durata della connessione, ricevere e scambiare messaggi e file con gli altri client connessi allo stesso server e sostituisce i caratteri ':' con il relativo smile.

2.2 Casi d'uso

I casi d'uso per questo progetto sono molteplici e differiscono in base all'attore:

- I casi d'uso del **client** sono:
 - setNickName
 - inviaMessaggio
 - riceviMessaggio
 - inviaFile
 - riceviFile
- I casi d'uso del **server** sono:
 - addUser
 - inoltraMessaggio
 - inoltraFile

Analizziamo nel dettaglio i casi d'uso del **client**.

Il caso d'uso **setNickName** permette all'utente di inserire un nome che lo distinguerà dagli altri utenti e che sarà valido fino alla sua disconnessione.

Il caso d'uso **inviaMessaggio** permette al client di inoltrare il messaggio che scrive al server e quest'ultimo si occuperà di distribuirlo agli altri client connessi. Questo include altri 2 metodi: **convertMessage** e **send**. Il primo si occupa della conversione del messaggio in un array di byte per poi poter esser inviato al server tramite metodo send.

Il caso d'uso **riceviMessaggio** si occupa di ricevere i messaggi che il server inoltra al client e di stamparli a video.

I casi d'uso **inviaFile** e **riceviFile** sono quasi identici a **inviaMessaggio** e **riceviMessaggio**: questi permettono di inviare al server e ricevere da questo un file. Il primo include 2 metodi: **convertFile** e **send** che si occupano di trasformare il file in modo da poter essere inviato al server.

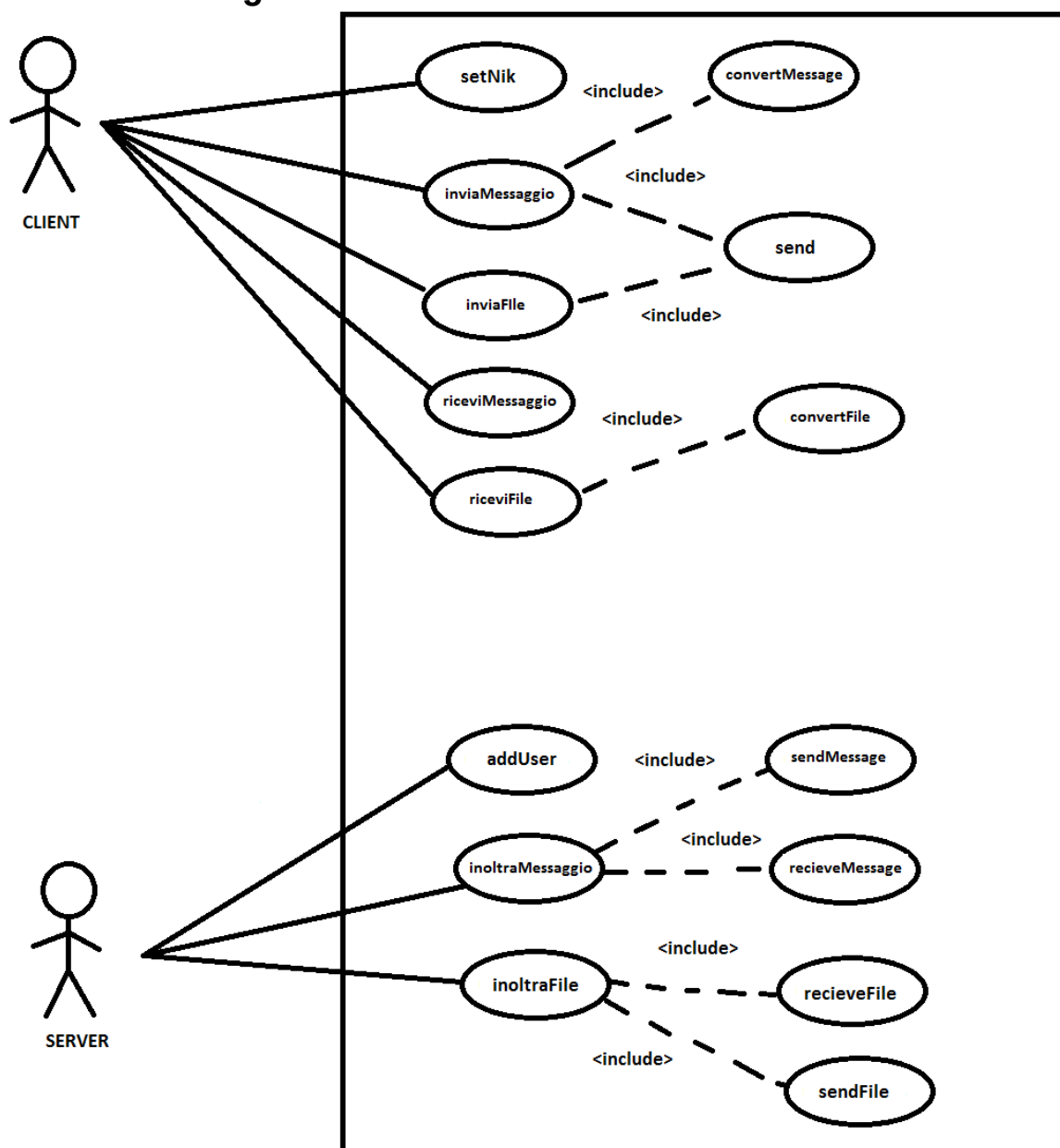
Ora prendiamo in esame i casi d'uso del **server**.

AddUser permette di aggiungere un client alla lista degli utenti a cui andranno inoltrati i messaggi provenienti dagli altri client.

InoltraMessaggio manda il messaggio che gli è arrivato da un client a tutti gli altri. Questo include il metodo **recieveMessage**, che serve per poter ricevere l'array di byte da client, e **sendMessage** che manda l'array a tutti gli altri client.

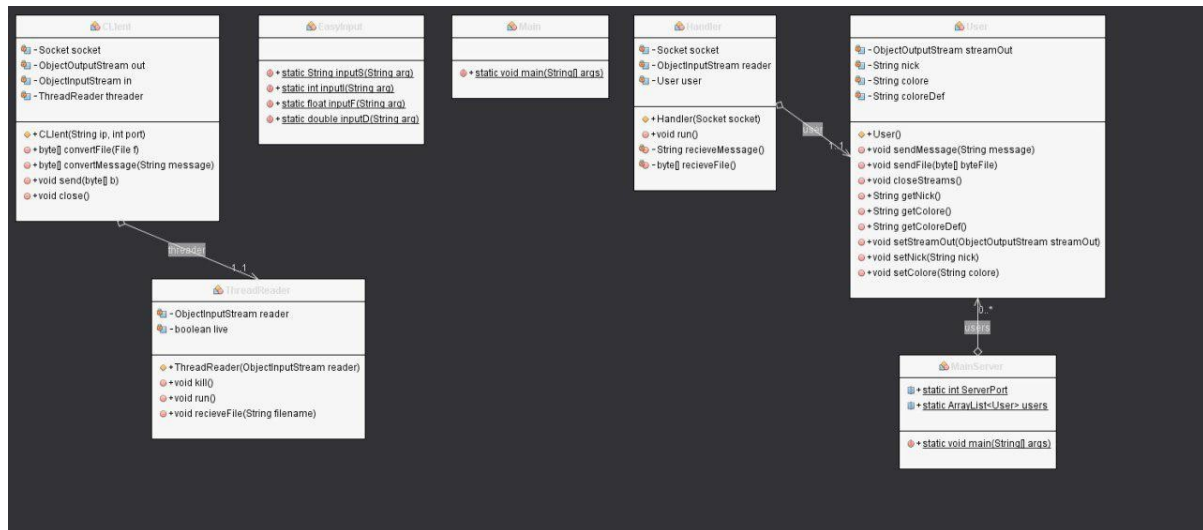
InoltraFile è identico al precedente solo che invece di ricevere e inoltrare un messaggio riceve e inoltra un file. Include **recieveFile** e **sendFile** che hanno la funzione di ricevere un file dal client e inoltrarlo agli altri utenti.

2.3 Diagramma dei casi d'uso



Documento di progettazione ed implementazione dell'applicazione

1. Diagramma delle classi



2. Commento del codice

Il JavaDoc è presente nella cartella javadoc/

3. Commento personale

Per la realizzazione di questa applicazione ho lavorato con il mio compagno di gruppo (Samuel Piccini) e abbiamo progettato insieme il lavoro che andava fatto, poi io ho iniziato lo sviluppo della “base” del progetto (ho implementato il modo in cui il server gestisce i vari client, come il client doveva inizialmente dialogare con il server, la gestione dei nickname ecc.) poi il mio compagno ha proseguito lo sviluppo adattando l'applicazione alle nuove esigenze (invio e ricezione di file, conversione del carattere ':' con uno smile, ecc). Ho realizzato il JavaDoc commentando nuovamente il progetto e insieme abbiamo redatto la relazione durante una chiamata skype (eccetto questa parte).