

Documentazione del progetto inerente alla realizzazione di una chat multiutente, sia per la parte del Server che del Client.

### Soluzione Utilizzata

È stato scelto di utilizzare il linguaggio Java per lo sviluppo dell'applicazione di messaggistica online. All'interno dell'applicazione, la grafica che permette l'accesso alle chat, lo scambio dei messaggi e gli eventi relativi all'utilizzo della stessa sono stati realizzati e implementati grazie alle componenti AWT e Swing che Java mette a disposizione. Per poter permettere al Server di ricevere le connessioni in entrata da parte dei Client è stata utilizzata la classe `ServerSocket`, che permette di aprire un servizio ad una determinata porta (in questo caso è stata scelta la porta numero 7777). Per permettere la comunicazione e lo scambio dei messaggi tra il Server e il Client è stata utilizzata la class `Socket` che permette di aprire un socket sapendo l'indirizzo del server e la porta sulla quale offre il servizio; il socket è uno stream che mette a disposizione un canale per l'invio e uno per la ricezione dei dati, e vengono utilizzati entrambi sia dal Client che dal Server. Per poter gestire la possibilità di avere più client connessi allo stesso Server, e di inviare e ricevere messaggi su ogni Client, si è ricorso all'utilizzo della programmazione multi-thread, implementata nell'applicazione grazie all'utilizzo delle classi `Thread` e `Runnable`: ad ogni nuova connessione al Server da parte di un Client viene istanziato un nuovo thread che gestisce la comunicazione da parte del suo unico Client, è infatti ogni istanza all'interno del Server che invia agli altri utenti i messaggi inviati dal suo Client; quando un Client accede alla chat online viene lanciata un nuovo thread che si occupa della ricezione dei messaggi, così da poter inviarli e riceverli allo stesso tempo. Un diverso thread, all'interno del Server, è stato utilizzato per poter permettere l'inserimento di particolari comandi quali: "list", "send", "close", "open", "exit" ed "—exit" le cui funzionalità saranno spiegate in seguito.

Errori e problematiche ancora da affrontare (nel documento sono già presenti sezioni di commenti contenenti informazioni sulle feature da aggiungere e improvement da correggere)

- Quando viene rimosso il pannello di una chat e viene effettuato il doppio click sul nome utente deve essere gestito cosa fare;
- Il mancato utilizzo di classi apposite per la gestione degli eventi rende meno leggibile il codice, ma un tentativo non è andato a buon fine;
- Il server potrebbe utilizzare un thread differente per l'invio dei messaggi, così da rimanere in ascolto dei messaggi inviati dal client, ma deve essere prestata attenzione a inviare i messaggi in ordine;
- Gestire la chiusura di un singolo server in caso di errore di comunicazione col proprio client.

## Applicazione lato Client

### Casi d'uso



#### Connessione

*Descrizione:* All'apertura dell'applicazione viene istanziata il socket per la connessione al server della chat;

*Considerazioni aggiuntive:* Essendo un'operazione statica non è richiesto alcun comando all'utente;

*Errori:* Il server non è attivo oppure è stato chiuso, viene chiusa l'applicazione;

*Risposta server:* Nessuna in caso di connessione effettuata, il programma continua.

#### Invio nominativo

*Considerazioni aggiuntive:* Viene data la possibilità al client di scegliere tra l'inserimento del nome utente desiderato oppure se farlo assegnare casualmente dal server. Dato che attualmente l'applicazione è

destinata ad un piccolo pubblico non è stato tenuto conto della limitazione sul numero di combinazioni possibili per i nomi utente;

*Risposta:* Dopo la risposta di positività della scelta del nome utente viene inviato a sua volta dal server al client.

#### **Scelto dall'utente**

*Descrizione:* L'utente inserisce il nome scelto;

*Errori:* Se il nominativo esiste già oppure non è del formato richiesto (lettere, numeri, underscore e trattino – da 3 a 20 caratteri);

*Risposta server:* un messaggio che ha per testo "OK" se il nome utente è valido, la stringa di errore se non lo è.

#### **Casuale**

*Descrizione:* L'utente spunta la casella della scelta;

*Risposta server:* un messaggio che ha per testo "OK".

### **Ricezione utenti connessi**

*Descrizione:* Una volta effettuato l'accesso il client rimane in attesa della lista dei nominativi degli utenti connessi;

*Considerazioni aggiuntive:* non è stato necessario uno scambio di messaggi per la richiesta dei nominativi dato che è un procedimento che viene effettuato all'inizio di ogni connessione;

*Risposta server:* La lista dei nominativi da inserire nel pannello degli utenti disponibili in chat.

### **Invio messaggio**

#### **Alla chat**

*Descrizione:* Viene inviato un messaggio ad un utente che può essere un altro client o una chat di gruppo, il destinatario viene scelto in base alla chat selezionata;

*Considerazioni aggiuntive:* Se non è stato scritto nulla il messaggio non viene inviato;

*Errori:* In caso di errata comunicazione col server;

*Risposta server:* Solo in caso di utente non raggiungibile o nel caso di solitudine all'interno della chat di gruppo.

#### **Di disconnessione**

*Descrizione:* Quando l'utente digita la stringa "fine" (all'interno della chat di gruppo) viene chiuso il processo che permette l'invio di messaggi;

*Errori:* In caso di errata comunicazione col server;

*Risposta server:* Un messaggio contenente nel testo la stringa "fine" che permette al client di chiudere l'applicazione.

#### **Di chiusura chat**

*Descrizione:* Quando l'utente digita la stringa "fine" (all'interno di una chat non di gruppo) viene chiuso il pannello di quella chat – il messaggio rimane all'interno del client;

*Considerazioni aggiuntive:* Ritenendo che l'utente non desideri più ricevere messaggi dalla chat con l'utente, anche alla ricezione di messaggi da esso la chat non viene riaperta.

### **Ricezione messaggio**

#### **Da parte di un client**

*Descrizione:* Il messaggio ricevuto viene inserito nella chat alla quale era destinato oppure nella chat privata col mittente di esso;

*Considerazioni aggiuntive:* Se la chat privata non esiste viene prima creata e poi inserito il messaggio.

#### **Di connessione**

*Descrizione:* Quando un nuovo utente si connette gli altri client ricevono il messaggio di connessione, e il nominativo viene aggiunto alla lista degli utenti connessi;

*Considerazioni aggiuntive:* Non è stato ritenuto necessario effettuare ad ogni nuova connessione (o disconnessione) o in modo periodico un fetch della lista completa degli utenti connessi.

### Di disconnessione

*Descrizione:* Quando un utente si disconnette gli altri client ricevono il messaggio di disconnessione, e il nominativo viene rimosso alla lista degli utenti connessi.

### Amministratore server

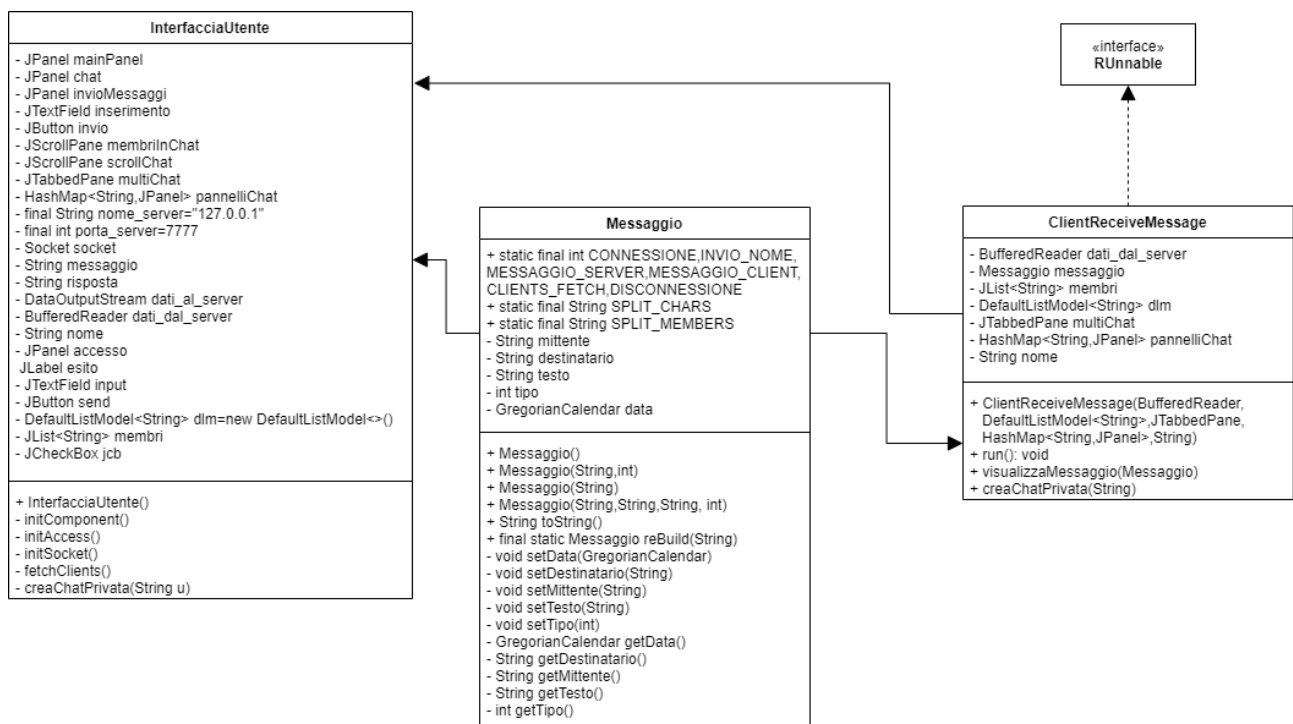
*Descrizione:* Quando viene ricevuto un messaggio dall'amministratore del server viene inserito nella chat globale;

*Considerazioni aggiuntive:* Entra a far parte dei messaggi da parte del server che sono visualizzati senza mittente.

### Chiusura server

*Descrizione:* Quando il server viene chiuso definitivamente ogni client riceve un messaggio avente nel testo la stringa "FINE" il che fa chiudere l'applicazione.

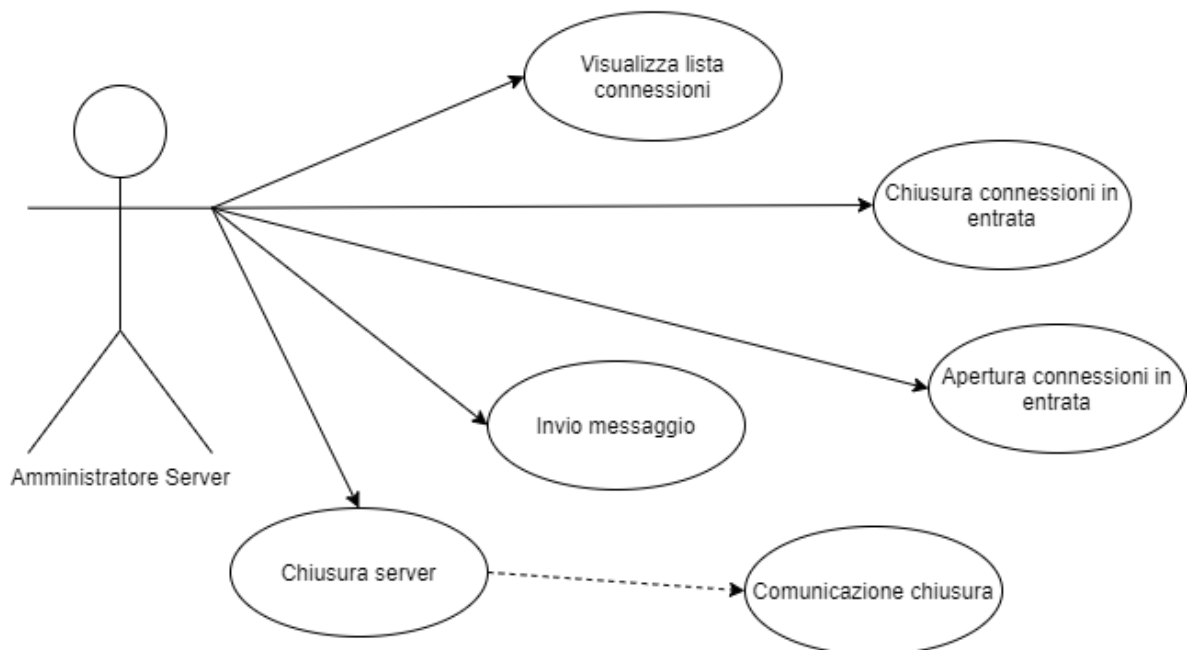
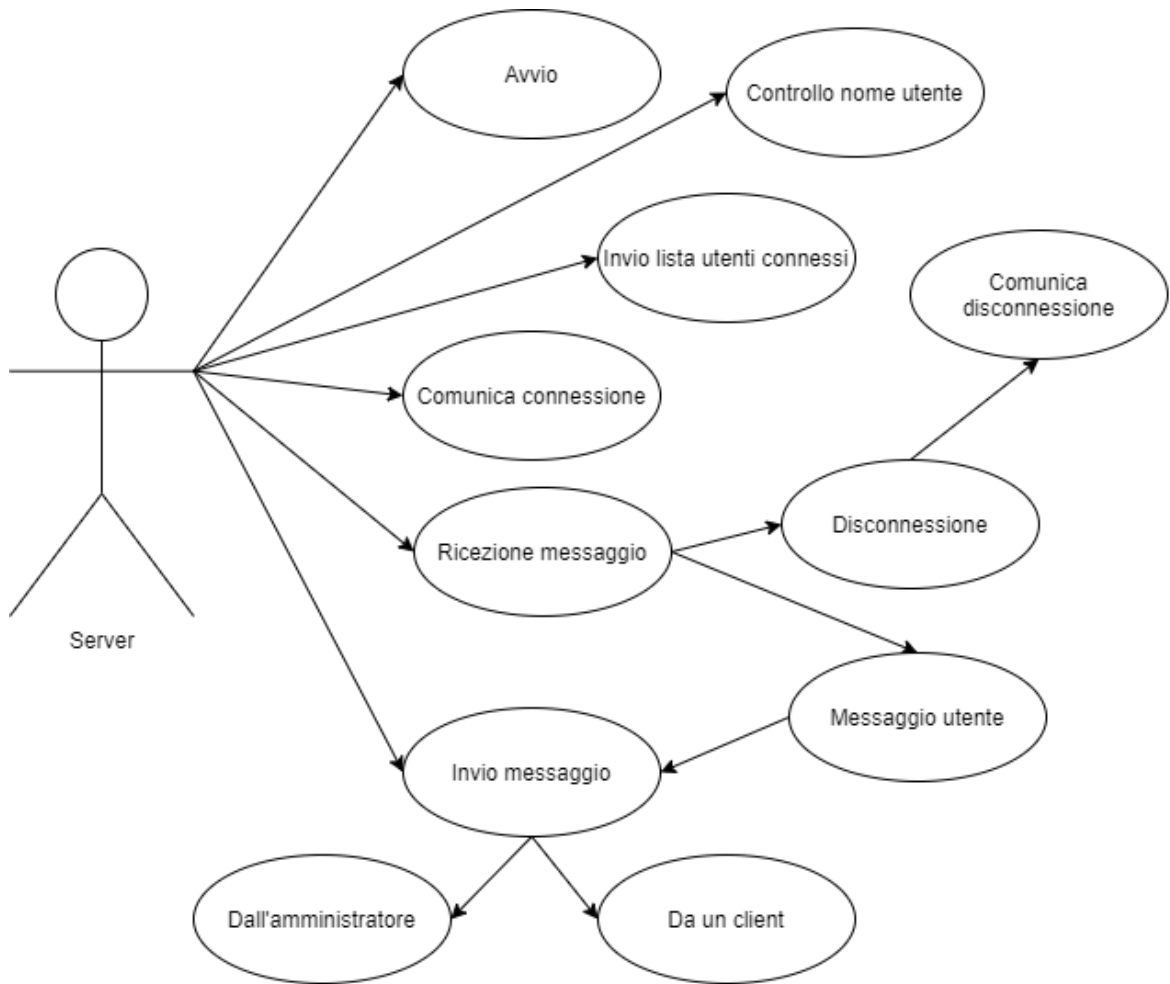
## Diagramma delle classi



La descrizione delle classi, dei metodi e delle loro funzionalità è inserita all'interno del codice sorgente e del javadoc; inoltre, si è cercato di dare dei nomi appropriati alle variabili e agli stessi metodi per comprenderne meglio gli utilizzi.

## Applicazione lato Server

### Casi d'uso



## Server

### **Avvio**

*Descrizione:* Il server si aggancia alla porta sulla quale offre il servizio, sta in attesa di una connessione e appena avviene crea un thread per essa;

*Errori:* Nel caso in cui la porta fosse già utilizzata;

*Considerazioni aggiuntive:* quando viene chiuso il server non viene chiusa l'applicazione, così da permettere di essere riaperto.

### **Controllo nome utente**

*Descrizione:* Il server controllo la correttezza e l'univocità del nome utente oppure ne genera uno casuale per il client;

*Risposta al client:* Invia l'esito del controllo e in caso fosse positivo invia anche il nome utente.

### **Invio lista utenti connessi**

*Descrizione:* Invia al client la lista degli utenti connessi al momento del collegamento, con al vertice una voce "Invia a tutti" per permettere di inviare in una volta un messaggio in privato ad ogni utente;

*Considerazioni aggiuntive:* Viene inviato anche se non vi fosse nessun utente connesso, in quanto necessario recapitare la voce "Invia a tutti".

### **Comunica connessione**

*Descrizione:* Il server comunica agli altri utenti connessi la riuscita connessione del suo client;

*Considerazioni aggiuntive:* Essendo il messaggio di tipo CONNESSIONE quando gli utenti lo ricevono inseriranno il suo nominativo nella lista degli utenti.

### **Ricezione messaggio**

#### **Disconnessione**

*Descrizione:* Quando riceve il messaggio scelto per la disconnessione del client rimuove l'utente dalla lista di quelli connessi.

Comunica disconnessione

*Descrizione:* Comunica agli altri utenti connessi la sua disconnessione;

*Considerazioni aggiuntive:* Essendo il messaggio di tipo DISCONNESSIONE quando gli utenti lo ricevono rimuoveranno il suo nominativo nella lista degli utenti connessi.

#### **Messaggio utente**

*Descrizione:* Quando riceve un messaggio di tipo MESSAGGIO\_CLIENT lo invia al destinatario scelto;

*Considerazioni aggiuntive:* Nel caso in cui fosse l'unico utente della chat lo comunica al proprio client, e nel caso in cui il destinatario si fosse disconnesso lo comunica lo stesso al client.

### **Invio messaggio**

#### **Da un client**

*Descrizione:* vedi la voce sopra descritta "Messaggio utente".

#### **Dall'amministratore**

*Descrizione:* Invia ad ogni utente connesso il messaggio scelto, con destinazione la chat globale.

## Amministratore Server

### **Visualizza lista connessioni**

*Descrizione:* Permette all'amministratore di visualizzare la lista degli utenti connessi con nominativo, indirizzo IP e porta del client;

*Considerazioni aggiuntive:* Visualizza un messaggio speciale nel caso in cui non vi fosse nessuno connesso.

### Chiusura connessioni in entrata

*Descrizione:* Permette all'amministratore di chiudere il socket del server così da non permettere ad altri utenti di connettersi, ma lascia invariata la comunicazione di coloro che sono già connessi;

*Considerazioni aggiuntive:* Se il server è già chiuso non viene effettuata nessuna chiusura.

### Apertura connessioni in entrata

*Descrizione:* Permette all'amministratore di aprire nuovamente il socket del server così da permettere ad altri utenti di connettersi alla stessa chat di coloro che già erano connessi;

*Considerazioni aggiuntive:* Se il server è già aperto non viene effettuata nessuna apertura.

### Invio messaggio

*Descrizione:* Permette all'amministratore di inviare un messaggio sulla chat globale a tutti gli utenti connessi.

### Chiusura server

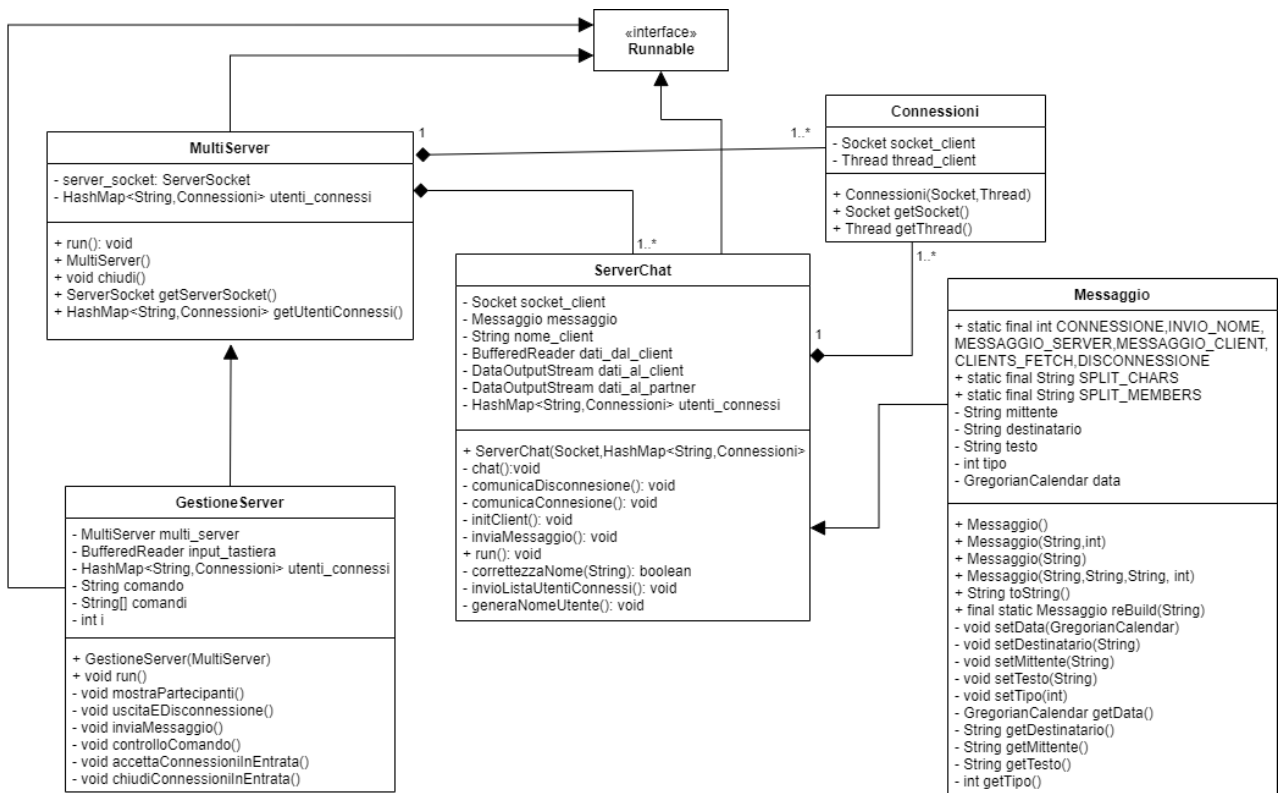
*Descrizione:* Chiude il server alle nuove connessioni e poi chiude anche l'applicazione del server.

#### Comunicazione chiusura

*Descrizione:* Può essere effettuata prima della chiusura dell'applicazione per comunicare l'imminente chiusura del server e serve a chiudere le connessioni aperte con i client;

*Considerazioni aggiuntive:* In caso di svariati utenti connessi potrebbe risultare un'operazione onerosa in termini di tempo, per questo è opzionale farla, dipende dall'urgenza della chiusura del server.

## Diagramma delle classi



La descrizione delle classi, dei metodi e delle loro funzionalità è inserita all'interno del codice sorgente e del javadoc; inoltre, si è cercato di dare dei nomi appropriati alle variabili e agli stessi metodi per comprenderne meglio gli utilizzi.



## Tipologie dei messaggi scambiati tra il Client e il Server

Tipo di messaggio	Campi richiesti	Fruitore	Descrizione
CONNESSIONE	Mittente	Il server	È il messaggio che permette al server di comunicare la connessione dell'utente agli altri componenti della chat.
INVIO_NOME	Testo	Il client	
MESSAGGIO_SERVER	Testo	Il server	Viene utilizzato per inviare comunicazioni al client da parte del server.
MESSAGGIO_CLIENT	Testo, Mittente e Destinatario	Il client	
CLIENTS_FETCH	\\	\\	Inutilizzato per la scelta della soluzione
DISCONNESSIONE	Mittente	Il server	È il messaggio che permette al server di comunicare la disconnessione (volontaria o involontaria) dell'utente agli altri componenti della chat.