## Documentazione

- ChatFase02 -

# Indice:

1.	Introduzione	(p.	3)
2.	Client	(p.	4)
3.	Server	(p.	8)
4.	Canale comunicaz	(p.	9)
5.	Database (1	<b>.</b>	10)
6	Diagramma casi uso (	$\widehat{}$	111

#### Introduzione:

ChatFase02 è un progetto scolastico volto alla realizzazione di una chat client to client (con possibilità di chattare in modalità multi-utente).

Il messaggio inviato dall'utente passa dal server che, senza leggerne il contenuto, si occupa di recapitarlo al/i corretto/i destinatario/i.

La realizzazione è avvenuta tramite il linguaggio di programmazione Java. Le classi di supporto utilizzate sono:

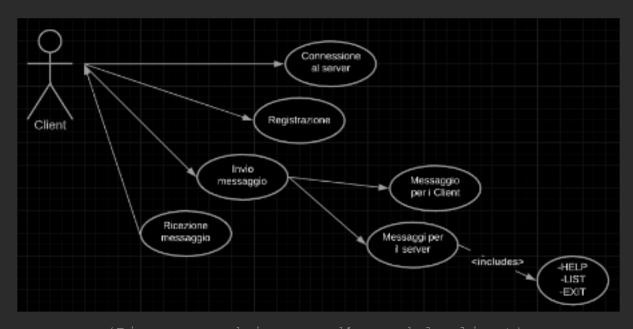
- Socket;
- Swing;
- Thread;

### Client:

Il Client è l'utente che usufruisce del servizio di chat offerto dal server.

Il client ha dunque la possibilità di iscriversi al server per poter scrivere e ricevere messaggi dagli altri utenti registrati.

Qualora l'utente non volesse più usufruire del servizio può eseguire il logout.



(Diagramma dei caso d'uso del client)

#### Metodi:

#### - connessione al server:

Per eseguire una connessione al server è necessario conoscere l'indirizzo ip e la porta del server. La classe Client si occuperà di questo automaticamente (senza chiedere all'utente a quale server si vuole connettere).

#### - registrazione:

Per registrarsi al servizio basta inserire un username univoco e valido (il server si occuperà di verificarne l'univocità, mentre la correttezza è compito del client).

#### - inserimento username:

L'inserimento dell'username rispetta una regex

 $("^[a-zA-z0-9._-]{3,20}$")$ .

(Se si necessita di verificare la validità di un username per conto proprio è possibile recarsi sul sito <a href="https://regexr.com/">https://regexr.com/</a> e testare la validità dell'username voluto)

### - invio del messaggio:

L'invio del messaggio necessita di un destinatario e di un corpo.

E' possibile scrivere un messaggio a più utenti semplicemente scrivendo i loro nomi nella sezione dei destinatari separati da una virgola e uno spazio (es: "MarioRossi, LuigiVerdi").

Lasciare lo spazio del destinatario vuoto comporta l'invio del messaggio a tutti. Per inviare un messaggio è necessario scriverlo nella sezione "messaggio" e poi premere il tasto invio sulla tastiera o il pulsante invio della GUI.

### - messaggi speciali:

I messaggi speciali sono i messaggi dedicati al server. Per inviare questi messaggi è necessario scrivere come destinatario solamente Server.

I messaggi possibili sono:

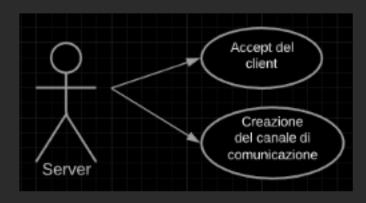
- "HELP": il server risponde con la lista dei comandi
- "LIST": il server risponde con la lista degli utenti connessi
- "EXIT": il server ti
  disconnette (in locale cessano i
  Thread di supporto al Client
  (ClientRecive, ClientSend)).

#### - ricezione dei messaggi:

I messaggi ricevuti vengono visualizzati nella sezione apposita e con il formato: "[NomeMittente]| [Messaggio]".

### Server:

Il Server si occupa di accettare un client e di creare un canale di comunicazione dedicato a lui.



(Diagramma dei caso d'uso del Server)

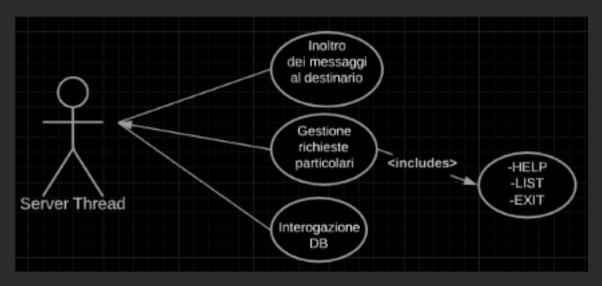
### Canale di comunicazione:

Il Canale di comunicazione si occupa di smistare ai destinatari i messaggi inviati dal suo client.

Il Canale di comunicazione si occupa anche di gestire i messaggi destinati al server offrendo la risposta al client.

Il Canale di comunicazione si occupa anche di eseguire il logout del client.

Il Canale di comunicazione si occupa anche di comunicare con il DataBase.

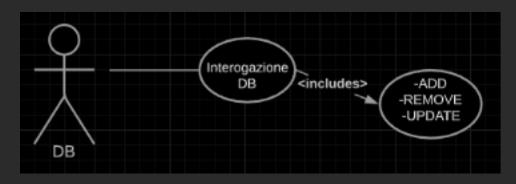


(Diagramma dei caso d'uso del Canale di comunicazione)

### Database:

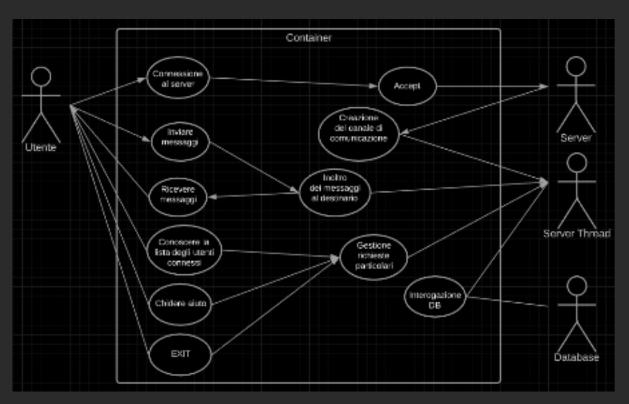
Il Database si occupa di contenere tutti gli utenti e i relativi socket.

Se interrogato correttamente il Database può aggiungere, rimuovere o modificare un utente.



(Diagramma dei caso d'uso del Database)

## Diagramma uso ChatFase02:



(Diagramma dei caso d'uso del progetto)