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

Classe 5AInf

a.s. 2016/2017

Relazione progetto

"TCP Chat"

Studente: Amedeo Di Gaetano

Docenti: Monica Ciuchetti, Luca Mencagli

Documento di Specifica dei Requisiti (Software Requirements Specification)

1. Introduzione e formulazione del problema

Il problema richiede di realizzare una chat punto-punto sincrona, utilizzando il protocollo TCP per la comunicazione. In particolare, il progetto sarà composto da due applicazioni, che dovranno interagire anche su dispositivi separati:

- Client, l'host che tenterà di instaurare una connessione;
- Server, la macchina che rimarrà in ascolto per ricevere richieste.

Inoltre, è necessario implementare le seguenti funzionalità:

- Scambio di messaggi ordinari;
- Visualizzazione su standard output;
- Differenziazione dell'autore di un messaggio secondo un codice colore;
- Possibilità di chiusura connessione in qualsiasi momento;
- Scambio di comandi, noti ad entrambi gli host, che permettano:
    - Di cambiare in proprio nome;
    - Di cambiare il proprio stato, scegliendo tra:
        - Disponibile;
        - Non Disponibile.
    - Di inviare uno smile;
    - Di inviare l'ultimo messaggio ricevuto;
    - Di terminare la conversazione.

Successivamente, si potranno implementare:

- La creazione di diverse chat simultaneamente, sfruttando diverse porte;
- La cifratura dei messaggi scambiati;
- Nuovi comandi, tra cui:
    - L'invio di una emoticon rappresentante il "like";
    - L'invio di un allegato;
    - Lo scambio di saluti.

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

2.1 Attori

L'applicazione prevede l'esistenza di due attori principali, che dovranno interagire tra loro simultaneamente, anche se in esecuzione su dispositivi differenti:

- Client, che avrà il compito di effettuare una richiesta di connessione;
- Server, che rimarrà in ascolto di richieste da parte del client;

Entrambi disporranno di funzioni che permetteranno di scambiare messaggi, emoticon, file e altri comandi, che risultano identici per entrambi gli host.

2.2 Casi d'uso

Caso d'uso: Connessione Client

ID: UC1

Attori: Client, Server

Precondizioni:

1. Il Server sia attivo e in ascolto;
2. Si conosce l'indirizzo IP del Server.

Sequenza degli Eventi:

1. Il client richiede di connettersi al Server;
2. Il Server avvia la connessione;
3. Il Client scambia messaggi col Server in modalità half-duplex;
4. Il Client e/o il Server chiedono di chiudere la connessione;
5. Viene chiusa la connessione.

Postcondizioni:

Il Client termina l'esecuzione.

Scenari alternativi:

Eccezione a): Il Server non è in ascolto.

1. Il Client non instaura una connessione;
2. Il programma termina con un messaggio di errore.

Eccezione b): Il Client non riesce a trovare il Server.

1. Il Client non instaura una connessione;
2. Il programma termina con un messaggio di errore.

Eccezione c): Il Client non riesce ad instaurare una connessione.

1. Il programma termina con un messaggio di errore.

Caso d'uso: Connessione Server

ID: UC1

Attori: Server, Client

Precondizioni:

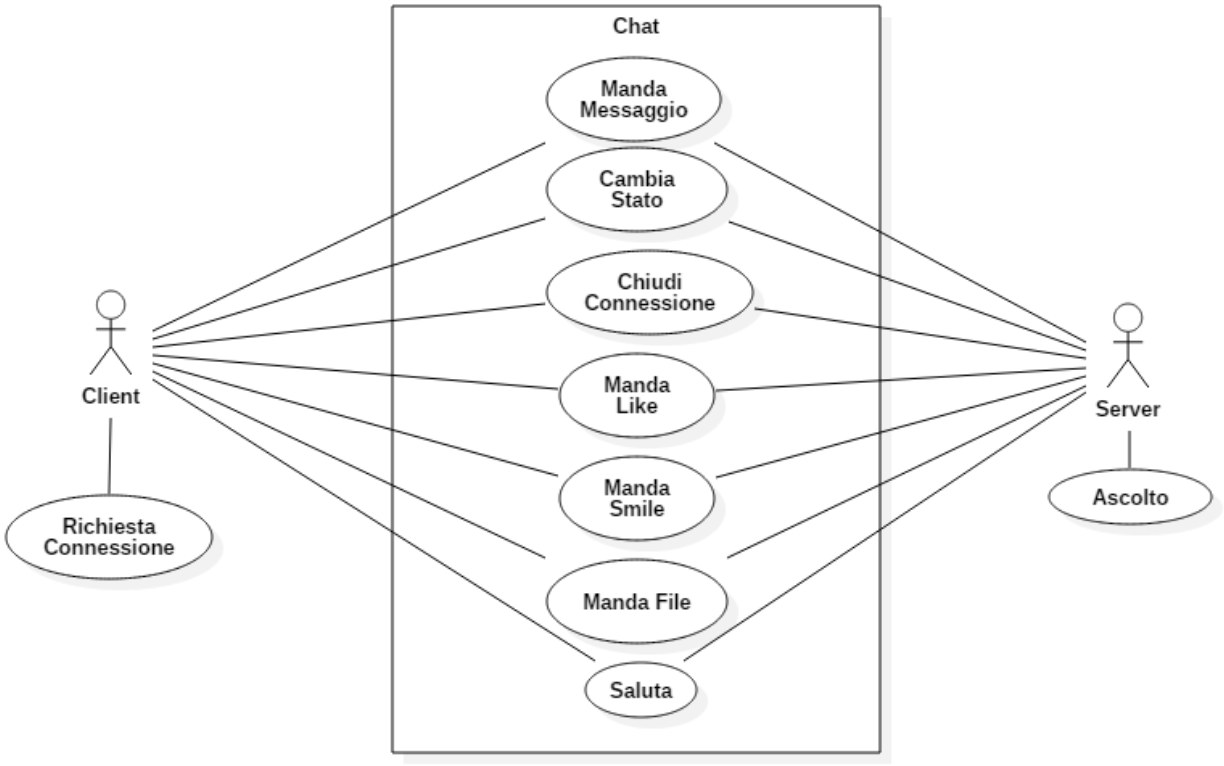
1. Ad un certo punto, un client chiede di iniziare una connessione

Sequenza degli Eventi:

1. Il Server si avvia e rimane in ascolto di richieste;
2. Il Server riceve una richiesta ed instaura una connessione;
3. Il Server scambia messaggi con il Client;

4. Il Server e/o il Client chiedono di chiudere la connessione; 5. Viene chiusa la connessione.
Postcondizioni: Il Server attende 10 secondi per eventuali altre connessioni.
Scenari alternativi: Eccezione a): Il Server non riceve richieste per almeno 10 secondi. 1. Il Server si interrompe. Eccezione b): Il Server non riesce ad eseguire il bind sulla porta. 1. Il Server termina con un messaggio di errore. Eccezione c): Il Server non riesce ad instaurare una connessione. 1. Il Server termina con un messaggio di errore.

2.3 Diagramma dei casi d'uso



2.4 Vincoli e tecnologie usate

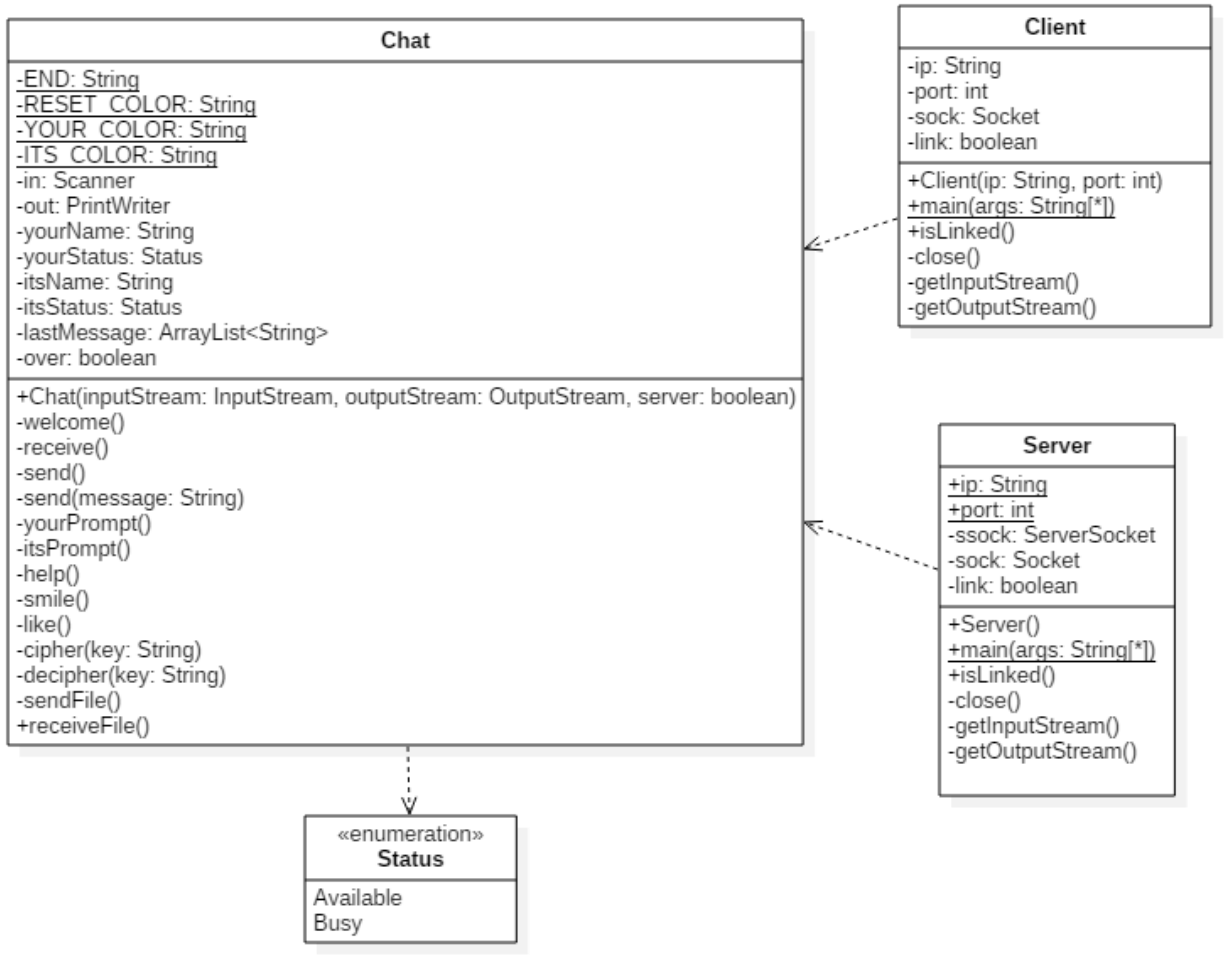
Per eseguire l'applicazione è necessario utilizzare un dispositivo che abbia installato il JRE di versione non minore alla 8.

2.5 Prototipo dell'interfaccia

L'interfaccia è a riga di comando, dove verranno visualizzati i prompt dei mittenti dei messaggi, preceduti da una lista iniziale dei comandi disponibili, comunque visualizzabile in qualsiasi momento.

Documento di progettazione ed implementazione dell'applicazione

1. Diagramma delle classi



2. Test dell'applicazione

Dopo aver avviato il processo server, che rimarrà in ascolto per 10 secondi, verrà instaurata una connessione col primo client che richiede la connessione. Dopodiché, i due processi si scambieranno messaggi in modo alternato, riconoscendo eventuali comandi speciali, i quali effettueranno una determinata operazione. Per chiudere la connessione verrà utilizzato l'apposito comando che, una volta inviato, provvederà a chiudere la connessione in modo sicuro.