

PyChat

Progetto di Programmazione di reti
Università di Bologna

Martin Tomassi; 0001077737
Francesco Pazzaglia; 0001077423
Luca Casadei; 0001069237

28 aprile 2024

Indice

1	Introduzione	2
2	Client	3
3	Server	4

Capitolo 1

Introduzione

Questa è la documentazione del progetto di Programmazione di Reti relativo alla Traccia 1: *Sistema di Chat Client-Server* in Python. Il progetto prevede la realizzazione di una chat che consente a più utenti di connettersi a un server centrale e comunicare tra loro in una chatroom condivisa.

Il progetto è composto da due componenti principali: il **client** e il **server**.

Il server gestisce le connessioni dei client e invia i messaggi in modalità broadcast all'interno della chat. Il client, invece, permette agli utenti di connettersi al server tramite protocollo TCP, di inviare messaggi e di visualizzare i messaggi ricevuti dalla chatroom degli altri utenti. Quando un client si connette al server, può vedere solo i messaggi inviati dopo che la connessione è stata stabilita; i messaggi precedenti non vengono "salvati" nel server.

E' stato scelto di utilizzare il protocollo a livello di trasporto chiamato **Transmission Control Protocol** perché garantisce la consegna affidabile dei dati, quindi nel caso di una chat è fondamentale che i messaggi vengano inviati e ricevuti in maniera corretta senza alcun tipo di perdita.

Capitolo 2

Client

Per stabilire una connessione, il client deve specificare il proprio indirizzo IP in rete e un nome utente univoco, che servirà a identificarlo all'interno della chatroom.

Inizialmente, il client crea un socket e tenta di stabilire una connessione con il server utilizzando l'indirizzo IP e la porta specificati. Se la connessione non riesce, il client gestisce l'errore sollevando un'eccezione.

Una volta stabilita la connessione, il client utilizza due thread separati per la trasmissione e la ricezione dei messaggi.

Durante la fase di trasmissione, il client acquisisce il messaggio dall'input del terminale e lo invia al server tramite il suo socket. Prima di inviare il messaggio, viene applicata la codifica UTF-8 per garantire la compatibilità con caratteri speciali come le lettere accentate.

Nella fase di ricezione, il client decodifica i messaggi ricevuti dal server, sempre utilizzando UTF-8, in modo da poter leggere correttamente i messaggi inviati dagli altri utenti della chatroom.

Capitolo 3

Server