

# “Relazione Chat Client-Server”

per il corso Programmazione di Reti  
A.A. 2023/2024

Elisa Yan

elisa.yan@studio.unibo.it

12 maggio 2024

## 1 Introduzione

Per il progetto di Programmazione di Reti è stato richiesto di sviluppare un sistema che implementa una chat client-server in Python utilizzando socket programming.

Il server deve essere in grado di gestire più client contemporaneamente e di consentire agli utenti di inviare e ricevere messaggi in una chatroom condivisa.

## 2 Design ed implementazione

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

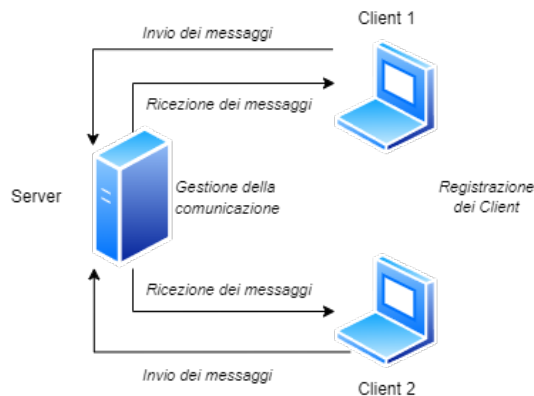


Figura 1: Funzionamento del sistema

### 2.1 Server

Il server si mette in ascolto su un socket TCP per accettare le connessioni dei client, garantendo una comunicazione stabile. Una volta che un client si è connesso, viene

richiesto l'inserimento di un nickname, associato su lui. Successivamente, viene inviato un messaggio a tutti i client connessi per notificare l'arrivo del nuovo utente. Il server funge da intermediario per i messaggi inviati dai client, inoltrandoli agli altri client connessi. In caso di disconnessione di un client, il server lo rimuove dalla lista degli utenti attivi e aggiorna l'elenco degli utenti. In sintesi, il server accetta le connessioni dei client, gestisce i loro nickname e facilita la trasmissione dei messaggi.

## 2.2 Client

Il client fornisce un'interfaccia grafica utilizzando la libreria **Tkinter** di Python, che include una casella di testo per l'inserimento dei messaggi e un'area per la visualizzazione dei messaggi della chat. Il client si connette al server specificando l'indirizzo IP e la porta del server. Una volta connesso, viene chiesto all'utente di inserire un nickname tramite una finestra di dialogo. Successivamente, viene avviata la GUI della chatroom, consentendo agli utenti di connettersi al server, inviare messaggi alla chatroom e ricevere messaggi dagli altri utenti. L'utente può digitare i messaggi nella casella di input e inviarli premendo il pulsante "Invia". I messaggi inviati vengono quindi codificati e inviati al server tramite il socket. Il client riceve i messaggi dal server e li visualizza nella finestra di chat.

## 3 Esempio di utilizzo

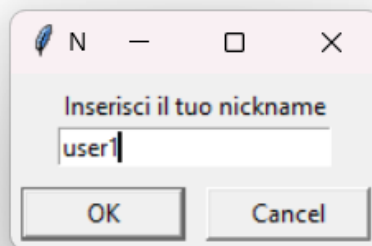


Figura 2: Inserimento del nickname

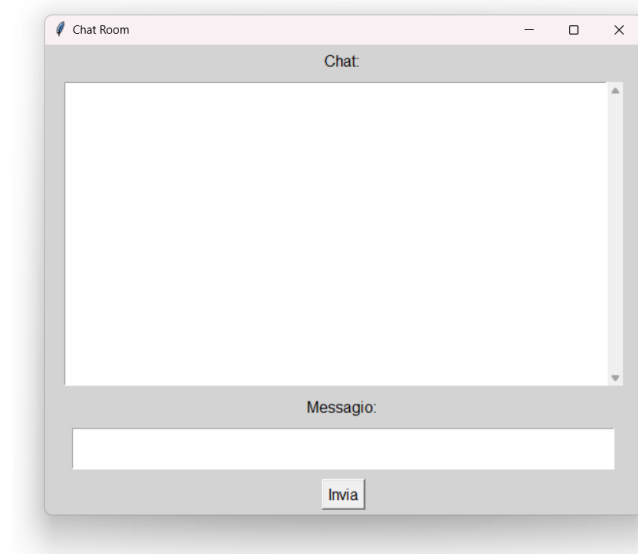


Figura 3: Entrata nella chat

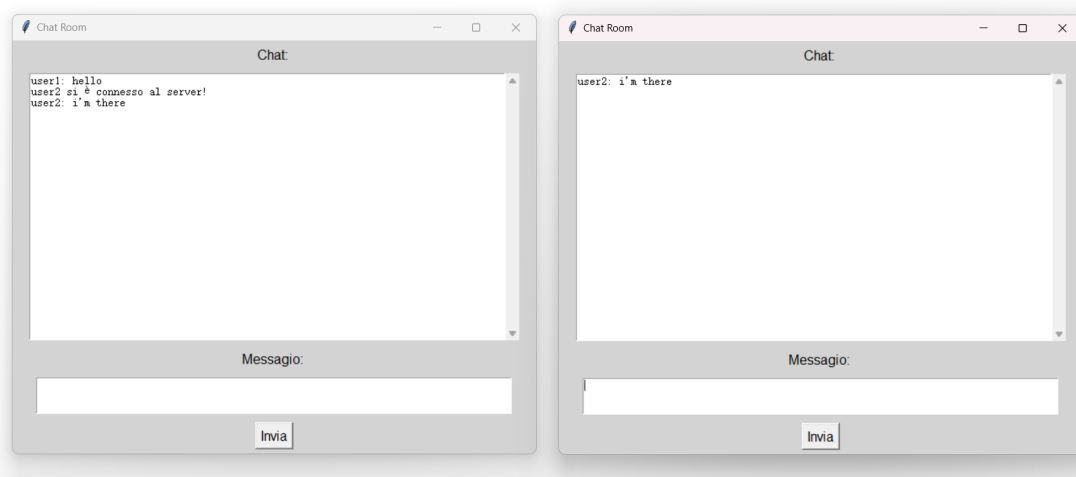


Figura 4: Entrata di un altro utente

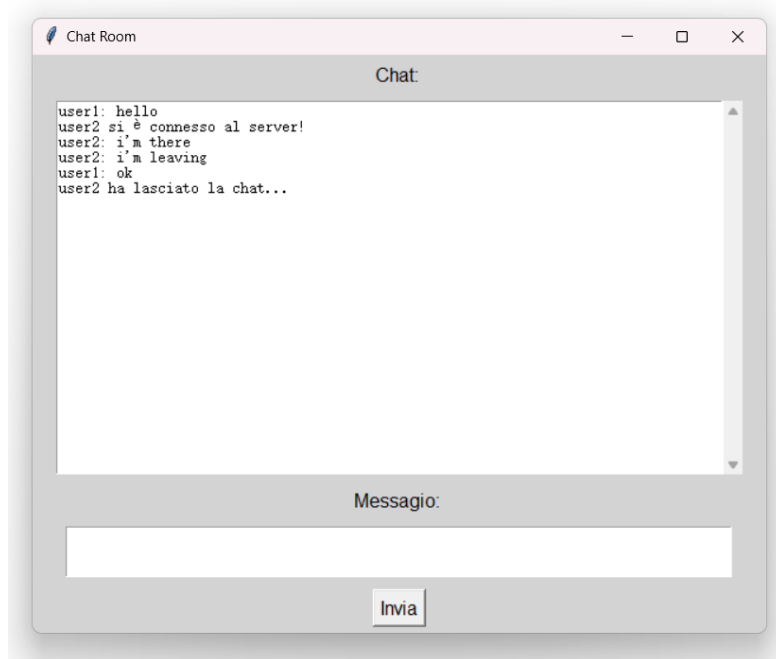


Figura 5: Uscita di un utente

## 4 Esecuzione dell'applicazione

Per eseguire l'applicazione del sistema di chat client-server, è necessario avere installato Python sul proprio sistema.

### 4.1 Server

Il server viene avviato digitando il comando sul terminale:

```
python .\server
```

Questo comando avvia il server, che si mette in ascolto dei client che cercano di connettersi.

### 4.2 Client

Il client viene avviato digitando il comando sul terminale:

```
python .\client
```

Durante l'esecuzione del client, all'utente viene richiesto di specificare l'indirizzo IP e la porta del server a cui desidera connettersi.

Una volta che il client si connette con successo al server, può inviare messaggi nella chat e ricevere messaggi dagli altri client connessi. La comunicazione tra client e server avviene tramite socket TCP/IP.

## **5 Commenti finali**

Il sistema attuale fornisce solo funzionalità di base, ma può essere esteso ulteriormente con nuove funzionalità e miglioramenti.