

Progetto per Programmazione di Reti Chat Server-Client

Questo progetto è composto da un file server che accetta richieste di più file client e ne gestisce la comunicazione di chat.

La repository contiene due file: (server.py) e (client.py).

Versione di Python.

Si è utilizzato python 3.11 per sviluppare questa applicazione.

Utilizzo di `server.py`

Si può far partire il server eseguendo:

```
$ python3 server.py
```

Fatto ciò il server chiede se si intende usare l'address di default (localhost con porta 50002, se si inviare S) oppure se si vuole digitare manualmente un address (inviare N).

```
C:\Users\Utente\Desktop\Uni\Progetto reti>python server.py
Utilizzare porta e indirizzo di default? (127.0.0.1,50002)? S/N
S
In ascolto per connessioni...
```

Se tutto va con successo, si potrà osservare che il server si mette in ascolto per connessioni. Da questo momento in poi il server è autonomo e scriverà da solo su console gli eventi principali (errori,disconnessioni,connessioni alla chat).

```
In ascolto per connessioni...
Accettata connessione da %s ('127.0.0.1', 64358)
Accettata connessione da %s ('127.0.0.1', 64363)
errore avvenuto per utente user2518,connessione interrotta
Impossibile raggiungere utente user2518,disconnessione in corso...
user2518 si è disconnesso
```

Utilizzo di `client.py`

Si può far partire il server eseguendo:

```
$ python3 client.py
```

Fatto ciò si apre un interfaccia grafica che permette all'utente di connettersi in 2 modi: -Fornendo username e un address (indirizzo e porta, cliccare il bottone prova a connetterti). -Fornendo username e codice di invito (cliccare il bottone prova a connetterti con codice).

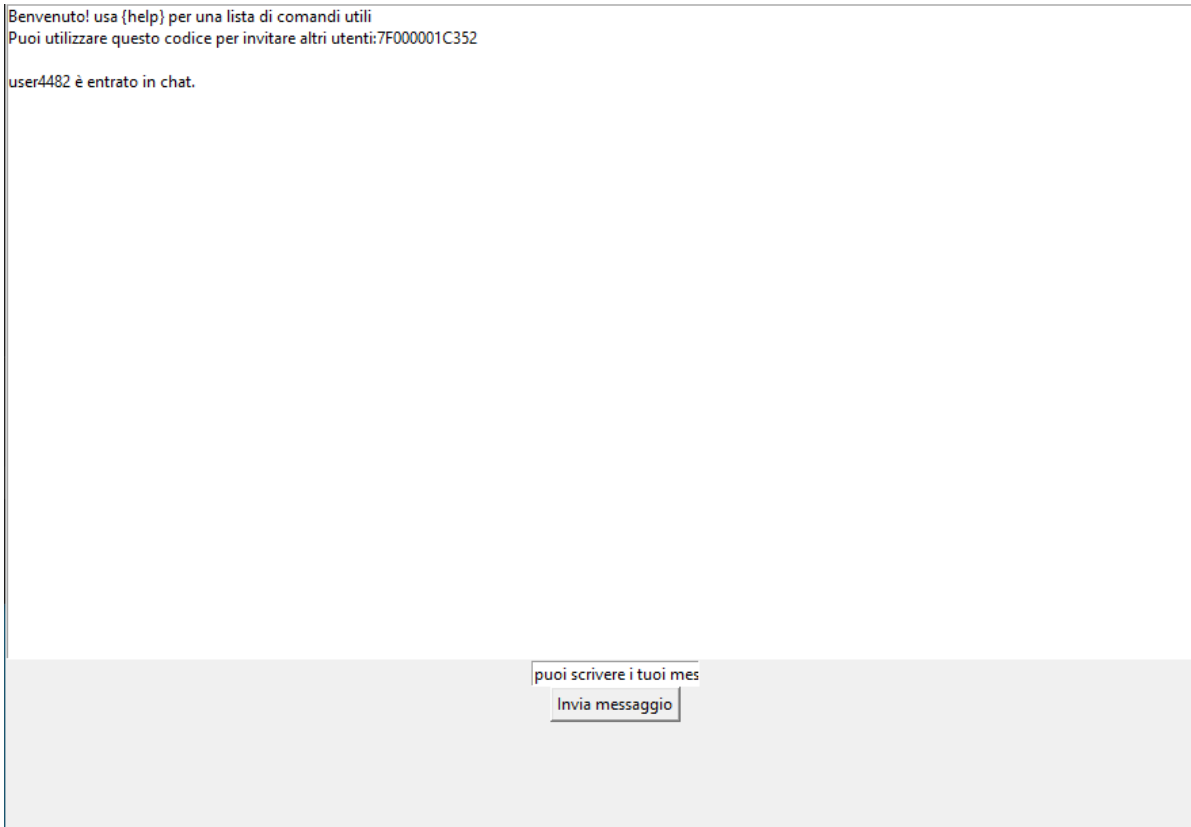
The screenshot shows a graphical user interface for a client application. It contains the following elements:

- A label "Fornisci uno username" above a text input field containing "user4482".
- A label "Indirizzo Ip" above a text input field containing "127.0.0.1".
- A label "Porta" above a text input field containing "50002".
- A button labeled "Prova a connetterti" below the IP and port fields.
- A label "Codice d'invito" above an empty text input field.
- A button labeled "Prova a connetterti con codice" below the invitation code field.
- A status label at the bottom that reads "nessun errore al momento!".

Il codice di invito deve essere dato da un altro utente già connesso in chat (viene fornito dalla chat stessa a inizio connessione).

Se in fase di connessione qualcosa va storto, una label riporta l'errore accaduto.

Una volta connessi con successo, la finestra mostrerà la chat con i suoi messaggi, e fornirà una casella di testo e un bottone di invio per permettere all'utente di scrivere i propri messaggi.



```
Benvenuto! usa {help} per una lista di comandi utili
Puoi utilizzare questo codice per invitare altri utenti:7F000001C352
user4482 è entrato in chat.
```

Si può uscire dalla chat:

- chiudendo la finestra.
- utilizzando il comando (quit) (scelta consigliata).

Errori comuni

1)Provando a connettersi,il client da "Connection refused": (client.py). Probabilmente il server non sta venendo trovato,assicurarsi di star usando codice/address corretto.

2)Il server non si avvia nonostante stia dando un indirizzo valido (server.py). Potrebbe essere che tale indirizzo sia già in uso,provare a cambiare la porta. Assicurarsi di non avere già un'istanza di server aperta.

Osservazioni tecniche.

1. Non vi è necessità di specificare parametri tramite linea di comando nè nel client perchè l'interfaccia grafica permette di eseguire tutto il necessario.
2. Questa applicazione utilizza solo librerie di default di python,non è necessario scaricare librerie aggiuntive.
3. Entrambe le parti cercano di gestire più eccezioni possibile per evitare una chiusura improvvisa del programma.

4. `server.py` impiega una classe `User` in modo da avere a portata di mano tutte le variabili importanti (`username, address` ecc...) in unico luogo.
5. Il codice di invito non è altro che un encoding in esadecimale di indirizzo ip e porta concatenati.