

## Server Chat UDP di Michael Berlingieri 5bi

(documentazione contenuta all'interno del progetto)

Link repository GitHub del progetto: <https://github.com/miberl/Chat-Server>

Indirizzo IP server remoto: 3.130.135.149

### RFC - protocollo stabilito:

Formato generale di un pacchetto:

```
-----  
| OPCODE | LENGTH | DATA |  
-----  
01 bytes  02 bytes  n bytes
```

#### 0. Messaggi generali di conferma/errore (Server -> Client )

0.1 'OK' -> Procedura completata correttamente

```
-----  
| 0 | 00 |  
-----
```

0.2 'Error' -> Procedura non completata causa errore

```
-----  
| 01 | * | ERROR_DESCRIPTION_STRING |  
-----
```

#### 1. Connessione iniziale - login

1.1 Registrazione (Client -> Server)

```
-----  
| 11 | LENGTH | USERNAME |  
-----
```

1.2 Risposta del server (Server -> Client)

'OK' ->     Procedura completata correttamente, utente aggiunto alla lista.  
           Pacchetto secondo il formato descritto in '0.1'

'Error' ->   Procedura non completata causa errore, utente non aggiunto alla lista.  
              Esempi errori: username già presente, lunghezza errata  
              Pacchetto secondo il formato descritto in '0.2'

## 2. Disconnessione - logout

### 2.1 Client manda richiesta di logout al server (Client -> Server)

```
-----  
| 12 | 00 |  
-----
```

### 2.2 Server conferma logout (Server -> Client)

'OK' ->     Procedura completata correttamente, utente rimosso dalla lista.  
           Pacchetto secondo il formato descritto in '0.1'

## 3. Informazioni

### 3.1 Client richiede informazioni generiche (ora e versione) al server (Client -> Server)

```
-----  
| 40 | 00 |  
-----
```

### 3.2 Risposta server (Server -> Client)

```
-----  
| 41 | LENGTH | SERVER_DATE_VERSION |  
-----
```

### 3.3 Richiesta lista utenti (Client -> Server)

```
-----  
| 42 | 00 |  
-----
```

### 3.4 Lista utenti registrati (Server -> Client)

```
-----  
| 43 | LENGTH | USER_1 | 0 | USER_2 | 0 | USER_N |  
-----
```

Byte 0 come separatore.

## 4. Messaggio pubblico

### 4.1 Client manda messaggio pubblico al server (Client\_mitt -> Server)

```
-----  
| 20 | LENGTH | MESSAGE |  
-----
```

### 4.2 Server inoltra il messaggio ricevuto a tutti gli utenti della lista (Server -> Users)

```
-----  
| 21 | * | SENDER | 0 | MESSAGE |  
-----
```

## 5. Messaggio privato

### 5.1 Client mittente invia il messaggio al server (Client\_m-> Server)

```
-----  
| 22 | * | RECEIVER | 0 | MESSAGE |  
-----
```

### 5.2 Server inoltra il messaggio all'utente specifico, se presente in lista (Server - Client\_d)

```
-----  
| 23 | * | SENDER | 0 | MESSAGE |  
-----
```

### Risposte (Server -> Client\_m)

'OK' ->   Procedura completata correttamente, destinatario presente il lista.  
          Pacchetto secondo il formato descritto in '0.1'

'Error' ->   Procedura non completata causa errore: utente non trovato.  
              Pacchetto secondo il formato descritto in '0.2'