

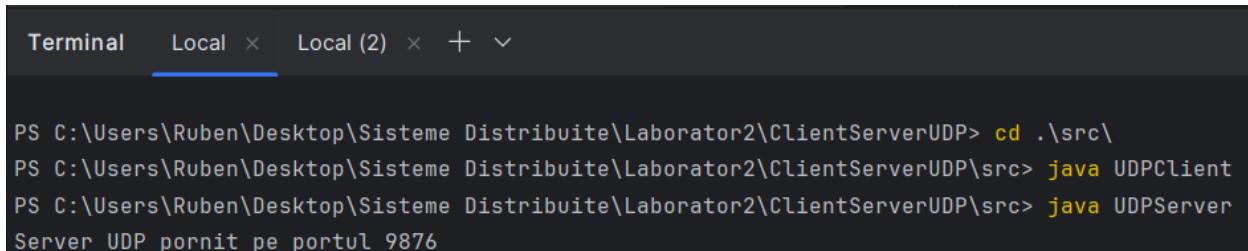
## Client Server UDP

Am create 2 clase: una pentru server si alta pentru client. În clasa server am utilizat un DatagramSocket pentru a crea un server UDP care ascultă pe portul 9876. Serverul folosește un buffer de 2048 de octeți pentru a primi pachete de date prin intermediul unui DatagramPacket. Programul rulează într-o buclă infinită și aşteptând să primească pachete de la clienți. Când se primește un pachet se extrag datele și lungimea lui, iar apoi serverul trimite înapoi același pachet către adresa și portul clientului care a trimis mesajul.

În clasa client am utilizat din nou DatagramSocket pentru a putea trimite și primi pachete UDP de la server. Am folosit o variabilă pentru a stoca adresa serverului(hard-codata) și alta pentru port. Am definit un array cu dimensiunile pachetelor (128, 512, 1024 bytes) apoi pentru fiecare dimensiune pentru fiecare dimensiune, se creează un pachet de date umplut cu octeți de 1. Se pornește cronometru, se trimit pachetul către server și apoi se aşteaptă răspunsul de la server. După ce sosește răspunsul, se măsoară timpul de răspuns (RTT) și se afișează rezultatul. Am testat programul folosind un PC din laborator pe care am rulat codul de client, iar pe laptopul personal codul de server. Mai jos sunt imagini cu rezultatele.

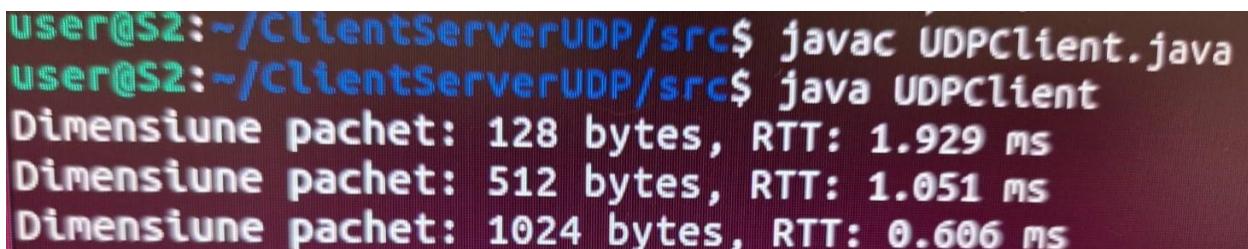
GitHub: <https://github.com/ruben-23/sisteme-distribuite/tree/main/Laborator2>

Server (Laptop Personal)



```
Terminal Local × Local (2) × + ▾
PS C:\Users\Ruben\Desktop\Sisteme Distribuite\Laborator2\ClientServerUDP> cd .\src\
PS C:\Users\Ruben\Desktop\Sisteme Distribuite\Laborator2\ClientServerUDP\src> java UDPClient
PS C:\Users\Ruben\Desktop\Sisteme Distribuite\Laborator2\ClientServerUDP\src> java UDPServer
Server UDP pornit pe portul 9876
```

Client (PC Laborator)



```
User@S2:~/ClientServerUDP/src$ javac UDPClient.java
user@S2:~/ClientServerUDP/src$ java UDPClient
Dimensiune pachet: 128 bytes, RTT: 1.929 ms
Dimensiune pachet: 512 bytes, RTT: 1.051 ms
Dimensiune pachet: 1024 bytes, RTT: 0.606 ms
```