# **Attività 1.1 (15 min):**

Analizza il seguente scenario e prova a prevedere cosa succede:

CLIENT



SERVER



**Cosa pensate che farà questo codice?**

Questo codice rappresenta una comunicazione UDP tra Client e Server, dove si scambiano dei messaggi in forma di pacchetti di Byte.

**Come si comporterà il server? E il client?**

I messaggi inviati e ricevuti vengono convertiti da Stringa a Byte e viceversa, il Server riceve il messaggio dal Client e converte la Stringa in maiuscolo e la rispedisce al Client in forma di Byte, per poi essere riconvertita in Stringa.

**Attività 2 (10 min):**

* Scarica i file presenti in FILE - PRIMM 4 (classroom);
* Esegui il codice;
* Quello che avevi scritto nell’attività 1, è confermato?

La run del codice conferma la descrizione precedentemente data

**Domande (15 min):**

1. **Quali classi Java sono usate per UDP?**

BufferedReader, InputStreamReader, DatagramPacket, DatagramSocket, InetAddress

1. **Come il client sa dove inviare il messaggio?**

Il Client sa dove inviare il messaggio inserendo l’IP e la porta del server

tramite InetAddress con la variabile IPServer.

1. **Perché usiamo byte[] e non direttamente String?**

Usiamo byte[] per poter avere una connessione UDP, perché per poter

fare ciò abbiamo la necessità di inviare dei Datagrammi (byte[]) e non

altri tipi di dato come Stringa, interi, ecc…

**Attività 3 (20 minuti)**

Modificare il codice per sperimentare:

1. **Aggiungere informazioni al messaggio del client:**

* Scopo: Rendere il messaggio più dettagliato.
* Esempio: Il client invia "Salve, sono il client X" invece del semplice testo.
* Cosa imparare: Concatenazione di stringhe, personalizzazione dell’output.

1. **Aggiungere un timestamp alla risposta del server:**

* Scopo: Restituire al client un messaggio arricchito con l’orario di elaborazione.
* Esempio: "CIAO [2024-12-01 14:30:45]".
* Cosa imparare: Utilizzo delle librerie Java per gestire data e ora.

1. **Gestire messaggi vuoti o invalidi:**

* Scopo: Evitare che il programma si blocchi o si comporti in modo imprevedibile.
* Esempio: Il server restituisce "Errore: messaggio vuoto!" se il client invia una stringa vuota.
* Cosa imparare: Implementazione di controlli condizionali per gestire input errati.

**Esercizio (20 min)**

1. Creare un’applicazione semplice in cui il client invia una stringa al server e riceve la lunghezza della stringa come risposta.
2. Creare un’applicazione semplice in cui il client invia una domanda (es. "Che ore sono?") e il server risponde con un messaggio appropriato (es. l'ora corrente).