

# Rapport

## Application réseaux

### Clavardamu

1] Le choix du langage est libre (le support sera fourni pour java, C ou python). Pensez à expliquer pourquoi vous avez retenu l'un plutôt que l'autre.

Nous avons choisis le langage java car c'est celui que nous avons utilisé lors des tp, il était donc plus facile de se lancer ayant un minimum d'expérience en java/réseau.

3.1] Comment attendre les messages de l'utilisateur et les transmettre au serveur et pouvoir afficher (sur le terminal) les messages reçus du serveur de manière concurrente ?

Pour que le client puisse attendre les entrées utilisateur tout en recevant les messages d'autres clients, nous avons utilisé des threads.

3.2] Type de serveur

Nous avons opté pour un serveur gérant un pool de thread dynamique

3.4.1] Proposer une architecture à base de files d'attente associées à chaque client

Nous avons essayé d'implémenter le système de file d'attente, mais si l'envoi de message fonctionnait bien, la réception n'était pas assurée.

3.4.2] Comment s'assurer que la communication est asynchrone ?

Les threads permettent d'assurer une communication asynchrone.