

Introducció

En aquesta pràctica es proposa la programació d'una aplicació Xat textual amb servidor centralitzat. El servidor fa un broadcast dels missatges originats per un client a la resta de clients. Cada client s'identifica per un **nick** que s'envia al servidor quan aquest es connecta.

El servidor manté un diccionari de parells (**nick, socket**). Aquest diccionari pot ser accedit en mode lectura o escriptura de forma que el servidor haurà de garantir-ne la seva consistència.

Parts de la pràctica

Es demana:

1. Programar dues classes **MySocket** i **MyServerSocket** que siguin funcionalment equivalents a les classes de Java **Socket** i **ServerSocket** però que encapsu-li'n excepcions i els corresponents streams de text **BufferedReader** i **PrintWriter**. Aquestes classes hauran de disposar de mètodes de lectura/escriptura dels tipus bàsics.
2. Programar el client com a dos threads concurrents, un que llegeix línies del teclat i les envia al servidor, i l'altre que llegeix línies del servidor—enviades per un altre client— i les imprimeix per pantalla. La plantilla és bàsicament la que es mostra a continuació:

```
// Input Thread
while ((línia = in.readLine()) != null)
    escriure línia per socket;

// Output Thread
while (hi ha línia del servidor)
    escriure línia per pantalla;
```

S'haurà de tenir en compte la forma d'acabar el client. **readLine** del teclat retorna **null** i tanca l'escriptura del socket. El fill del servidor que aten al client rep final de dades i acaba. Aquesta finalització en el servidor fa que s'acabin les línies a llegir per part del thread de sortida. No cal doncs cap dada de marca—metadata tipus **fi** per finalitzar l'execució.

3. Programar el servidor multithreading que encapsu-la un diccionari de parells (**nick, socket**). Per programar el diccionari feu servir la classe **Map**. S'ha d'accedir sincronitzadament al diccionari ja sigui en mode exclusiu o en mode lectura/escriptura. Hi ha diverses alternatives:
 1. Fer servir un monitor propi, d'exclusió mútua o lectors/escriptors.
 2. Fer servir la classe **ReentrantReadWriteLock**.
 3. Fer servir la classe **ConcurrentHashMap**.
 4. Fer servir el mètode **Collections.synchronizedMap**.

Francesc Oller