

## Activitat

AE2

## Ubicació

Tema 4

## Objectius

- Comprendre el concepte de socket.
- Aprendre a crear programes servidor i client.
- Saber crear connexions i comunicacions entre client i servidor.
- Aprendre a gestionar servidors multifil.

## Temporalització

La duració prevista per a esta activitat és de sis sessions lectives.

## Instruccions

Es demana crear una aplicació de tipus xat client-servidor multifil. Els requisits són:

Per a poder enviar missatges, el **client** s'ha d'autenticar primer. Per a fer-ho, quan es connecte al servidor li haurà d'enviar un usuari i una contrasenya. El servidor verificarà que usuari/contrasenya estan en un fitxer de text al qual només té accés el servidor. Si el client està autoritzat (i no està autenticat encara), tornarà un missatge "ok" i el client ja podrà enviar i rebre missatges. En cas contrari, tornarà un missatge d'error i el client haurà d'enviar-li de nou les credencials.

Com a restricció, el nom d'un usuari no pot contindre espais i ha de ser únic (no poden haver dos usuaris amb el mateix nom).

Una vegada autenticat, el client podrà enviar missatges al servidor. Estos missatges poden ser de 4 tipus:

- String "?": si el servidor rep aquest string, li tornarà al client un string que continga els noms de tots els clients que estan actualment connectats.
- String "@usuariX blablabla": en aquest cas, el contingut del missatge (blablabla) només s'enviarà a l'usuariX.
- String "blablabla": en aquest cas, el contingut del missatge (blablabla) s'envia a tots els usuaris connectats.
- String "exit": el client abandona el xat i tanca la connexió amb el servidor.

En paral·lel, el client podrà rebre missatges dels altres usuaris en qualsevol moment (sempre a través del servidor). Per tant, has de tindre en compte que el client haurà de tindre dos fils d'execució: un per a enviar missatges en qualsevol moment i un altre per a rebre missatges en qualsevol moment. Tots els missatges (per a enviar i rebuts) es mostraran en la mateixa consola. Cada missatge que s'envie o es reba ha d'anar acompanyat d'un timestamp quan aparega en la consola (el format del timestamp és lliure).

El **servidor** serà multifil i obrirà un fil per cada client que es connecte. Per a mantindre els clients correctament identificats, hauràs de crear una llista (ArrayList) amb els fils que hi ha connectats i que et servirà per a respondre a les peticions del client (saber quins clients hi ha connectats, enviar un missatge a un client concret o enviar un missatge a tots els clients). Quan un client es connecte s'afegirà a la llista i quan es desconnecte es llevarà de la llista. A més, cada vegada que crees un nou fil-servidor per a gestionar un client, a eixe fil hauràs de passar-li la llista amb tots els fils per a que pugui saber qui són i enviar-los missatges.

Cada vegada que el fil-servidor reba un missatge del client haurà d'identificar quin tipus de missatge és per a generar la resposta adequada i enviar-la a qui corresponga.

#### Suggerències (no són requisits de l'activitat):

Per a mostrar missatges propis o missatges d'altres clients pots alternar entre `System.out.println` i `System.err.println`, per a que canvie el color de les lletres en la consola.

El client emprarà la mateixa consola per a mostrar els missatges que envia i els que rep. Si has fet algunes proves de concurrència hauràs pogut comprovar que això pot fer que siga incòmode si es reben missatges quan s'està escrivint (es barreja tot en la consola). Una forma més neta de fer-ho és que els missatges que envia el client s'escriuen des d'una finestra popup. Per exemple, cada vegada que el client vulga enviar un missatge pot pressionar la tecla intro per a que aparega el popup.

Estructura suggerida per al projecte (pots plantejar-la d'altres formes):

- Aplicació **client**:

- Classe principal (amb mètode main):

Necessitarà un `BufferedReader` i un `PrintWriter` per a la part d'autenticació, i un `PrintWriter` (pots utilitzar el mateix) per a enviar els missatges al servidor i mostrar-los també a la consola del client.

- Classe auxiliar (fil, sense mètode main):

Necessitarà un `BufferedReader` per a rebre i mostrar a la consola del client els missatges del servidor (incloent els missatges que envien els altres clients).

- Aplicació **servidor**:

- Classe principal (amb mètode main):

Crea el `ServerSocket` i va admetent clients a través de fils, gestionant-los amb una llista, que passarà a cada fil que cree.

- Classe auxiliar (fil, un per cada client, sense mètode main):

Necessitarà un `BufferedReader` i un `PrintWriter` per a enviar/rebre missatges des del client al que està connectat i cap a cadascun dels clients que tindrà en la llista. Controlarà què enviar i a qui segons el missatge que li arribe del client.

#### **Avaluació**

L'activitat és obligatòria i es pot realitzar individualment o per parelles. Si es realitza per parelles, els dos alumnes han de fer l'entrega i indicar-ho clarament com a comentari amb l'entrega.

Per a l'avaluació es tindrà en compte el funcionament del programa, la codificació adequada i la documentació del mateix. També es sol·licitarà als alumnes que presenten i expliquen en classe l'aplicació que han desenvolupat.

El codi i la documentació s'haurà d'entregar a Florida Oberta (fitxer .zip) i també estar disponible al Github de l'alumne. S'ha d'entregar també un vídeo curt on es mostre l'execució de les funcionalitats desenvolupades.

## Recursos

Material del mòdul (Florida Oberta).

## Rúbrica

<b>Autenticació</b>	(0) No implementada.	(0,5) Falta controlar algun aspecte (per exemple, que el client ja estiga connectat).	(1) L'autenticació funciona correctament.
<b>Comunicació client-servidor</b>	(0) No implementada.	(1) Falta implementar algun/s tipus de missatge dels 4 que es demanen.	(2,5) S'implementen tots els missatges i es processen correctament al servidor.
<b>Comunicació servidor-client</b>	(0) No implementada.	(1) Envia missatges només al client que s'ha connectat i/o falten implementar funcionalitats segons el tipus de missatge del client.	(2,5) Envia missatges a tots els clients connectats segons les indicacions del missatge (a un o a tots els clients). El client rep i mostra correctament tots els missatges.
<b>Gestió de la llista de fils</b>	(0) No implementada.	(0,5) S'implementa parcialment o presenta errors (per exemple, no es passa correctament als fils o hi ha errors al recórrer-la).	(1) La llista de fils es gestiona correctament.
<b>Timestamp</b>	(0) No implementada.	(0,5) Falta implementar els timestamps per als missatges d'eixida o d'entrada.	(1) S'implementa correctament per a tots els missatges.
<b>Documentació</b>	(0) No generada.	(0,5) Falten mètodes per documentar i/o la descripció de classes i mètodes no és adequada i/o falta el vídeo explicatiu.	(1) La documentació està completa i detallada. El vídeo és es explicatiu.
<b>Qualitat del codi</b>	(0) No s'ha seguit la guia d'estil.	(0,5) El codi presenta alguns errors d'estil i/o alguns comentaris innecessaris.	(1) El codi és net, està ben estructurar i respon a la guia d'estil.