



EPS

Escola Politècnica
Superior

Projecte/Treball Fi de Carrera

Estudi: Eng. Tècn. Informàtica de Sistemes. Pla 2001

Títol: Implementació d'un sistema xat multiprotocol fent servir el llenguatge de programació ERLANG

Document: Resum P/TFC

Alumne: Walter Juan Pairó

Director/Tutor: Esteve del Acebo
Departament: Informàtica i Matemàtica Aplicada
Àrea: LSI

Convocatòria (mes/any): 09/2009

Resum P/TFC: Xat multiprotocol

Aquest projecte final de carreta d'Enginyeria Tècnica en Informàtica de Sistemes, ha consistit en la construcció d'una aplicació amb un llenguatge de programació concurrent i distribuït anomenat Erlang.

Erlang és un llenguatge de programació funcional amb avaluació estricta, és a dir, assignació única i que inclou una màquina virtual. Va ser dissenyat per la companyia Ericsson per realitzar aplicacions distribuïdes, tolerants a errors, execució en temps real i de funcionament ininterromput. Un dels grans avantatges és que proporciona un canvi de codi en calent de forma que aquest es pot canviar sense parar l'execució del mateix. Originalment Erlang era un llenguatge propietari d'Ericsson, però va ser cedit com a *open-source* l'any 1998.

Erlang el tractar-se d'un llenguatge espacialment desenvolupat per ser distribuït és molt útil en quan a aplicacions específiques per xarxes de comunicacions, també és molt fàcil la programació de funcions recursives i la creació i gestió de processos. Un procés no és res més que un programa informàtic en execució, i el paral·lelisme significa que es te més d'un procés que col·laboren junts per executar una tasca en concret.

L'aplicació desenvolupada ha estat la programació d'un xat multiprotocol, el qual s'ha realitzat una primera part que ha consistit en el construcció d'un servidor per la xarxa local i el seu corresponent client. Llavors per poder fer un client més funcional i útil s'ha implementat un altre protocol, IRC. Al tractar-se d'un llenguatge espacialment dissenyat per treballar en processos s'ha intentat dissenyar d'una manera la qual es pugés aprofitar aquesta qualitat.

Per tant l'aplicació queda dividida en dos parts, el client i el servidor. Per poder realitzar la comunicació entre el client i el servidor locals s'ha desenvolupat un protocol de comunicació local.

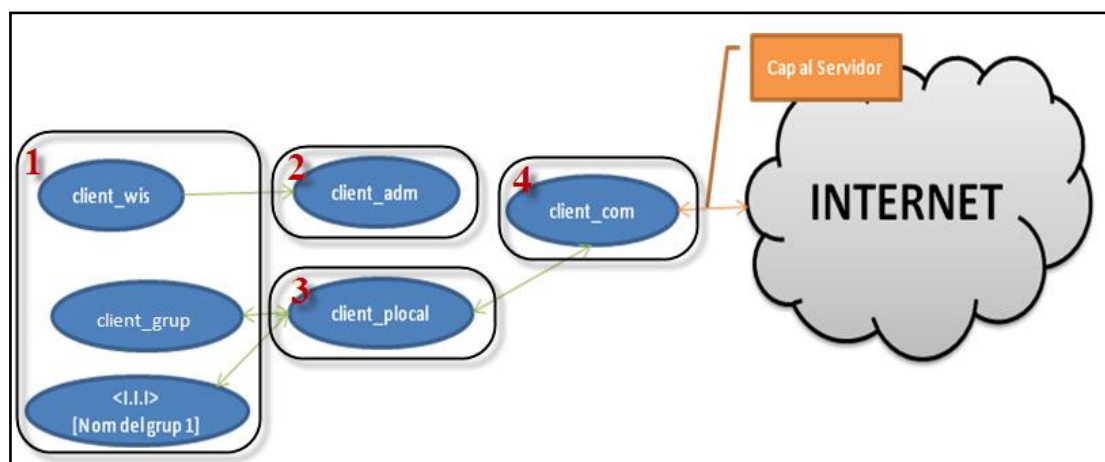
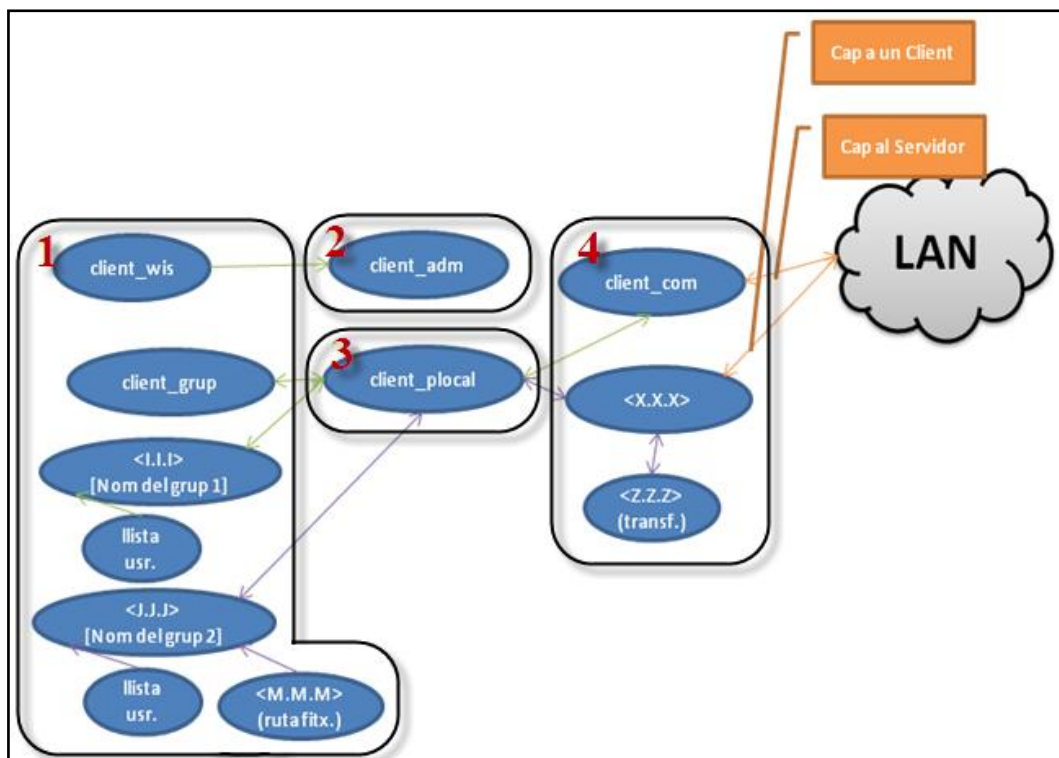
El client consta de diversos mòduls, el *client.eri*, *llista_usr.eri*, *miss_err.eri*, *pcomc.eri*, *pirc.eri*, *plocal.eri*, *wftp.eri* *wgr.eri* *winf.eri* i *wis.eri*. Es podria arribar a considerar que cada un d'aquest mòduls és el codi que necessita un procés específic per ser executat. A més a més el client consta d'una interfície gràfica pròpia, per això s'utilitza la llibreria *gs* que esta inclosa en les llibreries d'Erlang i això fa que aquesta aplicació sigui tan executable amb sistemes Linux com Windows.

El client en el protocol local permet realitzar en primer lloc una conversa tots a tots, llavors també es poden realitzar converses en grups determinats d'usuaris. Finalment també s'ha inclòs poder realitzar converses client a client per així no utilitzar tant el servidor, a més a més

amb les converses client a client es poden transmetre fitxers els quals queden guardat en una carpeta de *fitxers rebuts*.

Per altra banda el client en protocol local ofereix la possibilitat de connectar-se en un servidor IRC i entrar en un canal en concret. En el canal es poden realitzar converses client a client i la conversa tots a tots.

El disseny del client es podria dividir en 4 grans seccions que cada secció estaria formada per un o més processos. La primera (número 1 a la imatge) seria la que esta formada per tots els widgets (finestres, que interactuen amb l'usuari), el segons seria l'administrador encarregat de gestionar tots els errors i encarregar d'iniciar i finalitzar l'aplicació. El tercer grup estaria destinat al procés encarregat de descodificar el protocol que s'ha escollit, i per acabar el número quatre que és l'encarregat de comunicar-se amb l'exterior, la xarxa LAN o Internet. En les següents imatges es veuen els exemples en el protocol local i el IRC corresponentment:



Pel que fa el servidor local esta compost pels mòduls *grup.erl*, *miss_err.erl*, *pcoms.erl*, *servidor.erl* i *servidor_bdd.erl*. Al igual que el client cada un d'aquest mòduls es podria dir que contenen el codi necessari per executar un procés determinat, menys el *servidor_bdd.erl* ja que en aquest hi ha totes les funcions necessàries per engegar, parar, crear dades, eliminar-les o modificar-les de la base de dades. Aquesta base de dades ha estat desenvolupada amb Mnesia, una base de dades que es troba en les llibreries d'Erlang. La base de dades el servidor l'utilitza per la gestió del usuaris i dels grups de converses, per tant es creen i s'eliminen elements constantment.

El disseny del servidor varia una mica més que el del client ja que en el servidor no s'utilitza una interfície gràfica. Però igualment es pot dividir en grups, un primer grup (número 1) seria el format pel procés encarregat d'escoltar les peticions de connexió del client, llavors si la connexió és un èxit es crea un dels processos del grup número dos, els quals la seva funció és escoltar el socket per transmetre les dades en un altre grup. El número tres és el principal del servidor, és l'encarregat de iniciar-lo, parar-lo i gestionar el errors que es pugin trobar (similar al número dos del client). I per acabar hi ha el grup número quatre el qual és el que gestiona les converses en grup, aquí és on es troba la general i llavors es van creant les altre que els usuaris demanen. També no s'ha d'oblidar que el tercer grup, el format pel gestor també s'hi poden trobar "petits" processos que ell crea, aquests processos es creen segons la petició que demana el client ja que així s'augmenta el paral·lelisme, però aquest processos, són d'una duració determinada i curta per això no s'inclouen en la imatge que es mostra a continuació:

