Escalabilitat, usabilitat i ampliació de funcionalitats a la plataforma AXARM (Aplicació eXtensible per Assistència Remota i Monitorització)

Xavier Vallejo López

Màster en Informàtica Industrial i Automàtica

Universitat de Girona

Departament d'Arquitectura i Tecnologia de Computadors

Tutor: Antonio Bueno Delgado

21 de juliol de 2009

Agraïments

Al meu tutor Antonio Bueno Delgado per la seva inestimable ajuda a l'hora de dirigir i supervisar la meva feina.

A tots els integrants del projecte TRiEM que han treballat amb mi, i especialment a Shaila Jiménez, Willem Vanderhoydonk i Frederick van der Have per haver-me aguantat el dia a dia.

Als membres del grup de recerca BCDS (Broadband Communications and Distributed Systems) de la Universitat de Girona per resoldre'm tots els meus dubtes.

A la FEM (Fundació Esclerosi Múltiple) per mostrar el seu suport i implicació durant tot el projecte.

A la meva família per tot el seu recolzament i comprensió.

Índex

1 Introducció **1**

1.1 Telemedicina i eHealth 1

1.2 La malaltia: Esclerosi Múltiple 4

1.3 El projecte TRiEM 6

1.3.1 Descripció general del projecte 6

1.3.2 La proposta 8

1.3.3 Arquitectura de l'aplicació 9

1.3.4 Tecnologia de comunicació 11

1.3.5 Tecnologia multimèdia 12

2 Estat de l'art **13**

2.1 Situació de la telemedicina a Espanya 13

2.2 Experiència prèvia del projecte 18

2.3 Motivació del projecte 19

3 Servidors XMPP **21**

3.1 Descripció general 21

3.1.1 Openfire 22

3.1.2 ejabberd 23

3.1.3 Jabberd2 24

3.1.4 Tigase 24

3.2 Escollir un servidor XMPP pel projecte TRiEM 25

4 Usabilitat **26**

4.1 Millores en l'aplicació 26

4.1.1 Actualitzacions semi-automàtiques 26

4.1.1.1 Actualitzacions de les extensions 27

4.1.1.2 Actualitzacions del programa principal 28

4.1.2 Calendari d'activitats 29

4.2 Millores en el servei 30

4.2.1 Activitats asíncrones 30

4.2.1.1 Extensió memòria 31

4.2.1.2 Extensió carretera 32

4.2.1.3 Extensió macedònia 33

4.2.1.4 Extensió catifa de ball 34

4.2.1.5 Extensió figures 35

5 Seguretat: Autenticació **36**

5.1 Conceptes clau 36

5.1.1 Certificats X.509 i entitats certificadores CA 36

5.1.2 Xifratge TLS/SSL 38

5.1.3 Java Keystore 39

5.2 Per què fer servir X.509? 40

5.3 Integrar certificats en el servidor Openfire 40

5.4 Integrar certificats en els clients 42

5.5 Implementar l'ús de PKI en l'aplicació 43

5.6 Exemple d'ús 46

6 Seguretat: Confidencialitat **50**

6.1 Dades en dispositius extraïbles 51

6.2 Xifratge 53
6.2.1 Dades dels pacients amb TrueCrypt 53
6.2.2 Xifratge del servidor 55
6.3 Automuntatge de les dades xifrades 56

7 Escalabilitat **60**
7.1 Tsung: programa per estressar servidors 60
7.2 Experiments reals amb servidors XMPP 61
7.2.1 Preparació prèvia 61
7.2.2 Contingut del fitxer de configuració 63
7.2.3 Resultats 65
7.2.3.1 Màquina 1: OpenFire 66
7.2.3.2 Màquina 1: ejabberd 68
7.2.3.3 Màquina 2: Openfire 70
7.2.3.4 Màquina 2: ejabberd 72

8 SDK per crear una extensió **74**
8.1 Estructura d'una extensió 74
8.1.1 Propietats 74
8.1.2 Mètodes obligatoris 76
8.1.3 Events 77
8.2 Panell d'opcions propi 78
8.3 Funcionalitats comunes entre plugins 79
8.4 Implementar activitats asíncrones 80
8.4.1 Part de l'especialista 82
8.4.2 Part del pacient 83

9 Recursos externs **85**

10 Proves i resultats **89**

11 Conclusions **91**

12 Treball futur **93**

A Creació de Certificats amb OpenSSL **96**

B Fitxer configuració XML Tsung **101**

C Llistat dels components que fa servir AXARM **105**

D Test PC: Programa propi per llegir les característiques d'un PC **109**

D.1 Introducció 109

D.2 Requisits 110

D.3 Informació General 110

D.4 Desenvolupament del script 112

D.4.1 Què és AutoHotkey? 112

D.4.2 Fase I: Generació dades. msinfo32 113

D.4.3 Fase II: Compressió. Info-zip 114

D.4.4 Fase III: Enviament. Utilitat FTP Windows 114

D.5 Interacció amb l'usuari 116

D.5.1 Detalls tècnics 116

Bibliografia **117**

Índex de figures **122**

Capítol 1

Introducció

1.1 Telemedicina i eHealth

En tots els àmbits de la societat s'ha fet evident l'impacte de les TIC. Les TIC (Tecnologies de la Informació i Comunicació) són els camps que engloben tot allò referent a les noves tecnologies i, en general, al camp tecnològic. Amb la seva recent incorporació s'han fet grans avenços en diversos camps d'aplicació. El ventall de les TIC és enorme, però aquest projecte es centra dins del camp de la **telemedicina**.

La clàssica definició de telemedicina és la de la medicina a distància. Aquesta ja existia molt abans de l'aparició de les TIC fent servir el telèfon, la radiofonia o la televisió, encara que gràcies a les últimes novetats, aquest camp (i sobretot Internet) ha permès fer coses impensables com videoconferències o operacions de cirurgia a distància. Alguns experts consideren que es troba a mig camí entre la medicina convencional i la tecnologia. Una característica a destacar és que la telemedicina no és un substitut de la medicina convencional, sinó que s'ha de veure com un sistema de recolzament tant pels especialistes com pels pacients.

En l'actualitat, dins del camp de la telemedicina, s'està enfocant en una nova dimensió anomenada **eHealth**. Aquest terme d'origen anglosaxó significa literalment *salut electrònica*, és a dir, tot el que té cabuda dins el context de la telemedicina, però suportada per mitjans electrònics i de telecomunicació. La definició que dóna la OMS (Organització Mundial de la Salut) l'any 1997 diu que la telemedicina és el subministre de serveis d'atenció sanitària, en els casos en que la distància és el factor crític, portat per professionals que utilitzen les TIC per l'intercanvi d'informació vàlida per fer diagnòstics, prevenció i tractament de malalties. Aquests nous camps, a l'hora de millorar un servei que ja existeix, permeten possibilitats que d'una altra forma seria impossible. Per exemple, portar un seguiment mèdic a una tripulació que es troba en alta mar o a una expedició a l'Antàrtida.

Dins del camp de l'eHealth es defineixen unes metes a aconseguir [1]:

- Crear un entorn segur i fiable pels pacients, que es sentin còmodes amb ell i que notin que és útil.
- Fer participar els pacients activament en el procés de la seva teràpia, en aquest cas seria dins el camp de la telerehabilitació o telemonitorització.
- Oferir informació als usuaris. Aquesta informació pot ser de diferents temes (consells mèdics, informació general...).
  - A més a més, és necessari que aquesta informació sigui supervisada per experts mèdics (per verificar si és correcta).

  - Una mateixa informació pot anar dirigida a diferents grups d'usuaris (els dos grans grups són: pacients i especialistes). Cal que la informació s'adapti a cadascun d'aquests grups d'usuaris.

- Avisar als possibles usuaris dels riscs que comporta utilitzar Internet. Cal recordar, que Internet no és una xarxa segura a dia d'avui.

  - Amb l'aparició de nous dispositius, també cal disposar de noves possibilitats a l'hora de presentar la informació. Per exemple, una informació es pot presentar amb només text, o acompanyar-la d'imatges o vídeos segons l'ocasió.

Es podria incorporar un llistat dels múltiples camps que forma la telemedicina, cada un d'ells el suficientment ampli per exposar vàries tesis doctorals. No és objectiu d'aquest projecte explicar cada un d'aquests temes, i per tant ens centrarem en un en concret: **l'assistència sanitària a pacients amb malalties neurodegeneratives**.

Les malalties neurodegeneratives són totes aquelles que comporten un desordre cognitiu. Algunes d'aquestes malalties més conegudes són l'Alzheimer, el Parkinson, Creutzfeldt-Jakob i l'esclerosi múltiple. L'aplicació que presenta el projecte té com a objectiu oferir varis serveis a pacients amb **esclerosi múltiple:** tasques de rehabilitació, assistència remota o monitorització.

1.2 La malaltia: Esclerosi Múltiple

L'Esclerosi Múltiple (EM, també coneguda amb el nom *encephalomyelitis disseminata*) és una malaltia neurodegenerativa, crònica i no contagiosa que afecta al sistema central nerviós. Actua disminuint la mielina, una capa amb funcions aïllants que envolta a la fibra nerviosa de la neurona. Actualment no hi ha cap tractament per curar-la, però sí que hi ha medicaments per intentar alleugerir els seus efectes. Tampoc es saben les causes exactes que provoquen aquesta malaltia. A més a més, es presenten vàries variants d'esclerosi múltiple, la qual cosa fa que els pacients reaccionin de manera diferent al llarg dels anys.

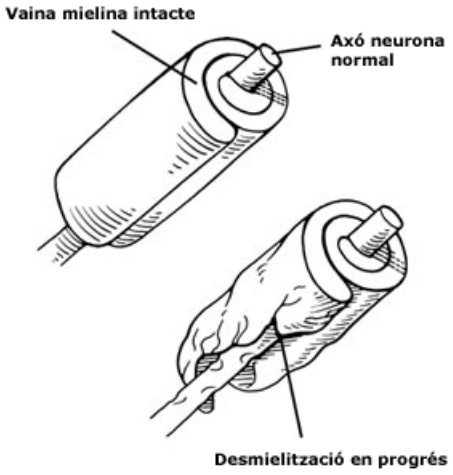


Figura 1.1: L'esclerosi múltiple afecta i danya al sistema nerviós del cos humà.

Un dels símptomes més freqüents de la malaltia és la reducció de la mobilitat del pacient que pot arribar fins a la invalidesa. Una dada esperançadora és que si és tractada, el 50% dels pacients recuperen bona part de la mobilitat, i només el 10% moren per aquesta causa. L'EM és la causa més freqüent de discapacitat neurològica no traumàtica en la població adulta jove. Afecta a 1 de cada 1000, en persones que solen estar entre 20-40 anys (mitjana de 32 anys), i afecta de 2-3 vegades més a dones que a homes.

L'EM també pot provocar deteriorament cognitiu a la persona. Entre un 43% i 65% dels pacients amb EM en tenen, i el més important és que molts cops és infradiagnosticat i s'etiqueta com una depressió, estrès o trastorns de personalitat.

Els símptomes típics dels pacients són: l'adormiment de les extremitats, espasmes, fatiga, dolor, cansament, alteracions en la vista... No necessàriament s'han de complir tots, ja que dependrà de cada pacient. Un dels principals problemes dels doctors és la dificultat d'assegurar el diagnòstic de la malaltia, ja que es sol confondre amb altres símptomes de menor importància, encara que en realitat l'EM ja es troba en el pacient. És clau poder detectar l'EM a temps, ja que el seu tractament immediat pot millorar molt la qualitat de vida del pacient.

Segons estudis recents, la prevalença de la EM és de 45-85 pacients cada 100.000 habitants, mentre que la incidència és de 5 nous pacients per 100.000 habitants/any. Cada any apareixen 324 nous pacients a Catalunya, dels quals 25 són gironins. Del nombre total de pacients amb EM, a Catalunya n'hi ha 4800, dels quals 350 són de la província de Girona.

1.3 El projecte TRiEM

1.3.1 Descripció general del projecte

Des de l'any 2005, el grup de recerca **BCDS (Broadband Communications and Distributed Systems)** [2] de la Universitat de Girona està treballant amb la **FEM (Fundació Esclerosi Múltiple)** [3] en un nou projecte anomenat TRiEM (TeleRehabilitació i Esclerosi Múltiple) per implementar una plataforma que permeti realitzar tasques de monitorització, teleassistència i telerehabilitació. Això permet ajudar als especialistes del centre a dur a terme activitats de rehabilitació a distància (el pacient a casa seva i l'especialista al centre).

La FEM és una entitat privada, sense afany de lucre, nascuda a Barcelona l'any 1989, amb la missió de millorar la qualitat de vida de les persones afectades d'esclerosi múltiple i destinar recursos a la investigació. Disposa d'un Hospital de Dia a cada capital de província, on els pacients hi van a fer rehabilitació (disposen de gimnàs, vestidors...). No és un hospital clàssic on s'operi o hi hagi metges, sinó que actua com un centre que disposa de varis especialistes. L'horari que segueixen és de 9 a 5 i només dies laborals, amb la qual cosa no poden respondre en un cas d'emergència.

Les característiques de l'esclerosi múltiple i les circumstàncies en les que treballen els centres de la FEM (econòmiques, geogràfiques, etc) fan molt desitjable poder realitzar remotament activitats de: consulta mèdica, fisioteràpia, psicologia, neu-

ropsicologia, logopèdia o teràpia ocupacional/recreativa. A més a més, el sistema contribueix a la qualitat de vida del pacient en poder fer les consultes des de la seva llar.

L'objectiu principal del projecte és desenvolupar una aplicació que permeti als especialistes visitar d'una manera virtual a casa dels pacients i minimitzar els desplaçaments a les consultes mèdiques, amb el doble objectiu d'oferir un seguiment més llarg i continu i alhora millorar la qualitat de vida dels pacients.

El sistema està dissenyat per fer servir, el més possible, una infraestructura estàndard de **baix cost**; tant en termes d'equipament informàtic (CPU, tarja gràfica, webcams) com en comunicacions. Per exemple: es pot oferir una comunicació d'àudio i vídeo acceptable a dues bandes amb una connexió ADSL o equivalent.

1.3.2 La proposta

A partir d'aquesta col·laboració entre les dues entitats, s'ha anat desenvolupant l'aplicació **AXARM** (Aplicació eXtensible per Assistència Remota i Monitorització). El programa facilita una eina molt útil als especialistes d'un centre per realitzar tasques de rehabilitació, assistència remota o monitorització amb pacients que es trobin en un altre punt físic a través d'Internet. En aquest punt, cal **diferenciar** entre el projecte TRiEM (tot el que engloba al treball entre especialistes i pacients) i l'aplicació AXARM (l'eina pròpiament dit).

- Aplicació de videoconferència amb possibilitat de gravació.
- Requeriments típics en una llar, pensats des d'un punt de vista de l'usuari.
- Interfície d'usuari fàcil de fer funcionar. Configuració simplificada.
- Extensible a noves **funcionalitats i perifèrics.**

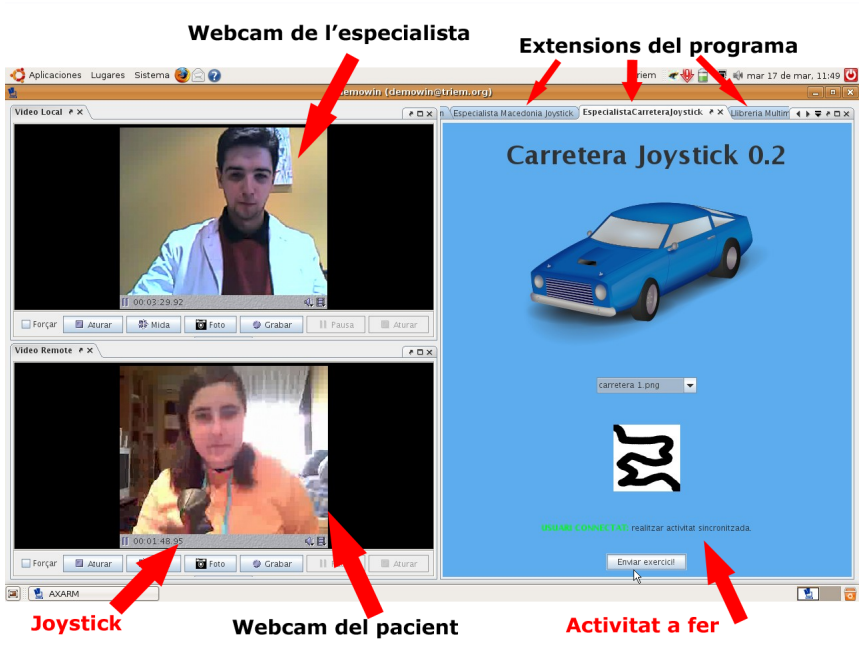


Figura 1.2: Imatge de l'aplicació en que es veu la interacció entre usuaris.

1.3.3 Arquitectura de l'aplicació

AXARM parteix de la base de JBother, un client de missatgeria instantània lliure[4] escrit en el llenguatge de programació **Java**. El principal avantatge del Java és que és un llenguatge multiplataforma (funciona en varis sistemes operatius que disposin d'una màquina virtual de Java) pel que facilita un ràpid desenvolupament de les necessitats que van apareixent.

Figura 1.3: Logotip del programa AXARM.

Tal i com s'ha esmentat en el punt anterior, una característica diferenciadora de l'aplicació està en les extensions. Es poden afegir fàcilment noves funcionalitats en l'aplicació sense haver de fer grans canvis.

Per simplificar les funcionalitats disponibles, es poden resumir en tres blocs:

A) Extensions **comunes** als pacients i especialistes:

- Videoconferència (tot i que l'especialista pot fer gravacions de les sessions i el pacient no).
- Una llibreria multimèdia on s'organitzen tots els fitxers que es generen (fotos, vídeos, resultats d'activitats...).
- Missatgeria instantània.

- Activitats entre especialistes i pacients:
  - Part de l'especialista: Permet generar estadístiques dels resultats que faci el pacient i guardar-les.
  - Part del pacient: Suport de perifèrics com *joysticks* o catifes de ball.

B) Extensions **exclusives de l'especialista**:

- Bloc de notes per l'especialista (funcionalitat equivalent al WordPad de Windows).

Per poder fer aquestes tasques, es fa servir una arquitectura *peer-to-peer* híbrida.

- L'arquitectura Client-Servidor es fa servir per enviar missatges de xat i missatges de control amb el servidor (autenticació, llista de contactes...).
- El P2P pur (connexió directa) es fa servir per transmetre les dades de la videoconferència entre usuaris. També es pot estendre a altres funcionalitats que requereixin una amplada de banda més gran que l'anterior.

Figura 1.4: Arquitectura híbrida utilitzada en el TRiEM.

1.3.4 Tecnologia de comunicació

L'aplicació funciona amb la tecnologia de comunicació XMPP[5] (*eXtensible Messaging and Presence Protocol,* conegut com *Jabber*). És un protocol de missatgeria instantània obert, documentat, estandarditzat i ampliable fent servir XML. Un exemple que el fa servir és *Google Talk*, el client de missatgeria instantània de Google.

• Descentralització: Qualsevol pot fer anar el seu propi servidor de XMPP i no hi ha un únic servidor central.

• Estàndards oberts: La IETF[6] ha formalitzat el XMPP com una tecnologia de missatgeria instantània (RFC 3920, RFC 3921)

• Seguretat: Els servidors XMPP disposen de sistemes de xifrat i autentificació segurs, com SASL i TLS/SSL.

• **Flexibilitat**: Funcionalitats pròpies que es poden construir a sobre del protocol.

L'últim punt és un dels més importants i que cal remarcar. El protocol XMPP és extensible (significa que un es pot construir el seu propi protocol a sobre del protocol de missatges XMPP). Gràcies a aquesta funcionalitat, es poden desenvolupar noves extensions i activitats que permetin enviar i rebre dades entre elles (es veuran quines activitats noves s'han afegit a l'aplicació en el capítol 4 secció 2).

Figura 1.5: Principals companyies que recolzen el XMPP.

1.3.5 Tecnologia multimèdia

Per transmetre la informació multimèdia (videoconferència), es fan servir les llibreries JMF de Sun Microsystems [7]. Actualment s'està considerant de fer servir una nova llibreria multimèdia (FMJ [8]) ja que JMF ha quedat desfasada (l'última actualització és del 2004).

• Protocol RTP i còdecs: Totes les dades multimèdia són enviades fent servir el protocol de streaming RTP a través de UDP, ja que resulta més important la fluïdesa de la videoconferència que no pas la seva qualitat. Amb UDP no hi ha retransmissions, i és més indicat per aquest tipus de tasca.

• El flux de vídeo és codificat fent servir el H.263 i l'àudio es codifica amb el format ULAW.

• H.263/RTP només transmet en: SQCIF (128x96), QCIF (176x144) i CIF (352x288). Encara que 352x288 no ofereix una bona resolució, l'experiència anterior ha demostrat que és suficient.

Una de les dificultats a l'hora de fer funcionar el programa a casa dels pacients, és que cal redirigir alguns ports en el router. Per a la videoconferència, calen 4 ports UDP oberts (del 4002 al 4005). El 4002 i 4003 són pel vídeo (4002 per dades, 4003 per control) i el 4004 i 4005 per l'àudio (4004 per dades, 4005 per control).

Capítol 2

Estat de l'art

2.1 Situació de la telemedicina a Espanya

En termes generals, l'abast de la telemedicina en el nostre estat és força pobre (considerant només la telemedicina que fa servir les noves TIC). Sí que hi ha un petit ús, però en molts casos es tracta de proves pilot o de grups reduïts. Per tant, encara queda molt de camí per recórrer per tal que s'ofereixi a un ampli sector de la societat.

Concretament a Catalunya, el **Departament de Salut** de la Generalitat de Catalunya disposa de l'AATRM [9] (Agència d'Avaluació de Tecnologia i Recerca Mèdica). L'AATRM és una empresa pública sense ànim de lucre adscrita al CatSalut (Servei Català de la Salut). La seva missió és proporcionar informació basada en el coneixement científic i en l'anàlisi del context sanitari, amb l'objectiu final de promoure que la introducció, l'adopció, la difusió i la utilització de les tecnologies mèdiques es faci d'acord amb criteris d'eficàcia, seguretat, efectivitat i eficiència demostrades científicament.

Una de les publicacions de l'AATRM que cal destacar [10] es centra en un estudi de la viabilitat d'un programa de telerehabilitació per pacients amb gran discapacitat d'origen neurològic (entre els quals, s'inclouen els pacients amb Esclerosi Múltiple). Es va publicar l'any 2007 per encàrrec del *Ministerio de Sanidad y Consumo*, i es presenten propostes de tot l'estat espanyol. No detallarem cada iniciativa que s'exposa, però sí que les enumerarem a continuació:

- **Telemedicina per l'atenció domiciliària.**
  - Pacients diabètics → Sistema Diabtel, Grup de Biomedicina i Telemedicina de la Universitat Politècnica de Madrid.
  - Pacients respiratoris → Plataforma CHRONIC, GBT-UPM i Hospital Clínic de Barcelona.
  - Pacients amb VIH/SIDA → Projecte Hospital VIHrtual, GBT-UPM i Hospital Clínic de Barcelona.
  - Pacients renals → TENHMS *(Trans-European Network initiative Home-Care Management System)*.
  - Pacients cardíacs → Projecte Airmed-Cardio, Hospital Puerta de Hierro de Madrid, Institut Carlos III i Fundació Vodafone Espanya.
- **Telerehabilitació motora.**
  - Extremitats superiors: Projectes H-CAD i HELLO-DOC (estudi clínic del 3/2005 al 2/2007 fet en: Fundació Privada Institut de Neurorrehabilitació Guttmann (Espanya), U.O. Riabilitazione Intensiva Neuromotoria (Itàlia) i National Multiple Sclerosis Center (Bèlgica))
- **Telerehabilitació cognitiva.**
  - Projecte EuroPaNet, (2001) Institut de Neurorehabilitació Guttmann.

Dels sistemes exposats, el que més s'acosta a la idea del TRiEM és el projecte **EuroPaNet**, que també consisteix en fer sessions de videoconferència i la realització d'activitats via correu electrònic. Es fa servir per oferir una prolongació del tractament un cop el pacient és donat d'alta, ajudant a la seva rehabilitació.

El projecte TRiEM no apareix en l'informe de la AATRM encara que en un futur pròxim pot aparèixer. Hi ha tres avantatges molt importants a destacar de la resta dels projectes observats:

- L'aplicació **AXARM no només serveix per pacients amb EM**. Encara que es centra en aquest col·lectiu, la modularització de l'aplicació permet adaptar-se a altres pacients. A més a més, el cost d'instal·lació és molt baix.

- Existeixen uns quants projectes que treballen extremitats superiors, però en canvi gairebé no n'hi ha per extremitats inferiors. AXARM disposa d'activitats per **treballar extremitats superiors i inferiors.**

- Respecte a les tasques ordinàries que es fan a l'Hospital de Dia, AXARM permet disposar de noves funcionalitats que no es podrien realitzar d'una altra manera. Per exemple, es poden programar activitats als pacients que les poden realitzar quan els hi vingui bé, alhora que es manté un control sobre aquests.

Altres iniciatives de la telemedicina també estan començant a néixer en la pròpia Universitat de Girona. Una possible evolució del projecte TRiEM resultaria en el projecte ARMAND *(Assistència Remota per MAlalties NeuroDegeneratives)*. En aquest cas es tractarien les consultes de telemedicina entre centres especialitzats i centres d'atenció primària. Finalment, destaquem en el nous estudis de grau de medicina la incorporació d'una assignatura optativa de telemedicina.

Transversalment als projectes de telemedicina i telerehabilitació, destaquem el saló AVANTE [11] celebrat a Fira Barcelona entre el 5 al 7 de Juny de 2008.

Figura 2.1: Logotip de la fira AVANTE.

Gràcies a la FEM, disposem d'un petit resum d'algunes de les ponències que es van realitzar durant el saló i que les resumirem a continuació:

1. **Institut Guttmann, Teresa Roig**

   Gràcies a la telerehabilitació s'eviten despeses, guanyen accessibilitat i alhora es va monitoritzant el progrés de l'usuari i poden intercanviar les dades amb altres especialistes. Van presentar els seus dos projectes:

   (a) EUROPANET (comentat anteriorment)

   (b) PREVIRNEC: És una plataforma per rehabilitació de seqüeles cognitives associades al dany cerebral. Disposa de varis programes d'exercicis per crear plans terapèutics personalitzats.

2. **Carlos III, José Maunal Sánchez i Alberto Jardón**

   En aquesta ocasió van explicar que disposen d'un centre d'innovació tecnològica per discapacitats i dependents.

   (a) Sistema de visionat de subtítols (3r millor invent de l'any)

   (b) Robot escalador ASIBOT (transportable que s'integra a la cadira de rodes)

3. **Gimnàstica i Realitat Virtual aplicada a la salut, Pedro Álvarez**

   Va explicar que era el terme GYMVR (Gimnàstica i Realitat Virtual) i de com crear teràpies de Realitat Virtual per curar el sedentarisme. Amb la utilització

de polseres RFID *(Radio Frequency IDentification)* es poden obtenir totes les dades de la persona (posició, pulsacions...). El gimnàs es troba en Argentona (a prop de Mataró) i es pot visitar per comprovar el seu funcionament.

4. **Temes d'investigació Smart Media, Santi Pérez**

   En aquesta conferència es va centrar en explicar materials intel·ligents, els quals permeten funcionalitats com: textures electroluminiscents, objectes que desprenen calor, que puguin emmagatzemar energia del cos humà i dels RFID. També va comentar que treballen amb l'Institut Guttmann amb la monitorització dels pacients durant un procés de rehabilitació. Per això, utilitzen sensors de moviment en elements ortopèdics.

Per acabar aquesta secció, mencionar que recentment s'ha donat a conèixer un nou projecte anomenat **Projecte EM-line!**. Consisteix en un *kit de treball* perquè els malalts d'esclerosi múltiple (en la fase inicial de la malaltia) puguin fer **rehabilitació cognitiva** a casa. Es tracta d'uns materials que permeten estimular i exercitar la memòria dels pacients als quals han diagnosticat recentment la malaltia. D'aquesta manera, es comença a tractar el pacient d'una manera senzilla que li permet continuar amb la seva vida diària sense interferències. Des del maig s'està realitzant un assaig clínic amb 44 pacients a l'hospital Josep Trueta.

El Projecte EM-line! compta amb la participació del servei de Neurologia de l'Hospital Josep Trueta de Girona, l'Institut d'Informàtica i Aplicacions de la Universitat de Girona i el servei de Ressonància Magnètica de la Clínica Girona.

2.2 Experiència prèvia del projecte

AXARM parteix de la base inicial del projecte TRiEM (TeleRehabilitació i Esclerosi Múltiple, FEM/UdG, 2005-06). Dins del marc acadèmic del projecte, s'han realitzat a la UdG dos projectes finals de carrera d'Enginyeria Informàtica (Carles Guadall, 2007 [12], i Xavier Vallejo, 2008 [13]), a més a més d'haver-se publicat dos **articles** que van estar acceptats als congressos eMediSys 2007 [14] i eHealth 2008 [15].

Ja en la fase inicial del TRiEM, es va realitzar una primera prova pilot amb cinc pacients amb EM. En l'actualitat, l'aplicació AXARM es fa servir en una altra prova pilot i es troba instal·lada a casa de sis pacients (alguns d'ells són els mateixos que els de la primera prova).

L'aplicació AXARM es va presentar en la **14a Edició dels Premis Patronat EPS a Projectes Finals de Carrera**, i va rebre el ***Premi al millor Projecte Final de Carrera d'Enginyeria Informàtica***, i una distinció especial de ***Reconeixement a l'Aplicabilitat Pràctica del millor Projecte Final de Carrera***. Gràcies a la col·laboració del Patronat també es va elaborar un **pòster** sobre el projecte TRiEM que es va exposar a l'edifici P-I juntament amb els altres participants.

El projecte TRiEM ha aparegut en altres publicacions, com per exemple en la revista Engega[16] de la pròpia Universitat de Girona.

Referent al marc econòmic, el projecte va estar finançat inicialment per la Fundació la Caixa (2005-06), i ha rebut dues beques destinades al seu desenvolupament (Beca-Col·laboració del MEC al 2007 i Beca BTI de la UdG al 2008). A més a més, la farmacèutica Novartis ha mostrat interès i està esponsoritzant el projecte durant 2009.

2.3 Motivació del projecte

En la fase més recent del TRiEM [13] es va descriure el funcionament de l'aplicació AXARM, mostrant tot el procés de refactorització que es va realitzar i, en conseqüència, les noves característiques afegides respecte al primer projecte del TRiEM [12]. En aquell moment es va centrar exclusivament en la part del client (part bàsica del sistema). En canvi, tota la part dedicada a la **configuració dels servidors** es va deixar de banda, expressament, conscients que s'hauria de reprendre més endavant. En conseqüència, amb l'experiència demostrada i consolidada dels treballs anteriors, la motivació d'aquest projecte és aprofundir en aquesta part.

La tasca principal d'aquest projecte és investigar i millorar el tema de la seguretat però sense deixar de banda la usabilitat ni tampoc l'escabilitat.

La **seguretat** explora diversos camps: des del xifratge de les dades locals i en les comunicacions fins a un sistema segur d'autenticació. Alhora, també es considera la no repudiació, molt útil per oferir un registre del control dels pacients.

Una de les màximes de la informàtica, és que mai es pot disposar d'un sistema 100% segur. En moltes ocasions, la seguretat es veu compromesa amb la usabilitat i viceversa. És impossible oferir un sistema molt segur i alhora molt accessible: cal un equilibri entre els dos. Pensant que l'aplicació l'ha de manejar un grup heterogeni de persones, sempre s'ha procurar oferir la màxima usabilitat possible i alhora un bon nivell de seguretat. Si un pacient concret vol disposar de més seguretat, se li poden oferir vàries solucions possibles.

Un aspecte important a tractar és el tema de les **dades personals** dels pacients. Durant una sessió normal de l'aplicació, es van generant continguts com són les con-

verses, resultats d'activitats, fotos o vídeos de les webcams. Totes aquestes dades s'han de tractar de manera sensible perquè són dades mèdiques, encara que no tenen tanta importància com, per exemple, els historials mèdics.

Continuant amb més aspectes del servidor, també s'ha considerat **l'escalabilitat** del sistema. Tots els servidors populars de missatgeria instantània XMPP asseguren que poden aguantar milers d'usuaris en bones condicions. Fins i tot es poden establir clústers de servidors XMPP per balancejar la càrrega d'aquests. Encara que en el context del projecte no s'arribarà als milers d'usuaris a mitjà termini, hem volgut provar l'eficàcia real dels servidors i els hem sotmès a una bateria de tests per avaluar-los.

Finalment, durant el transcurs del projecte també s'han millorat la usabilitat de l'aplicació: des de millores en la interfície del programa, fins a una millora del servei ofert, com per exemple que els especialistes poden enviar exercicis als pacients sense que estiguin *online*.

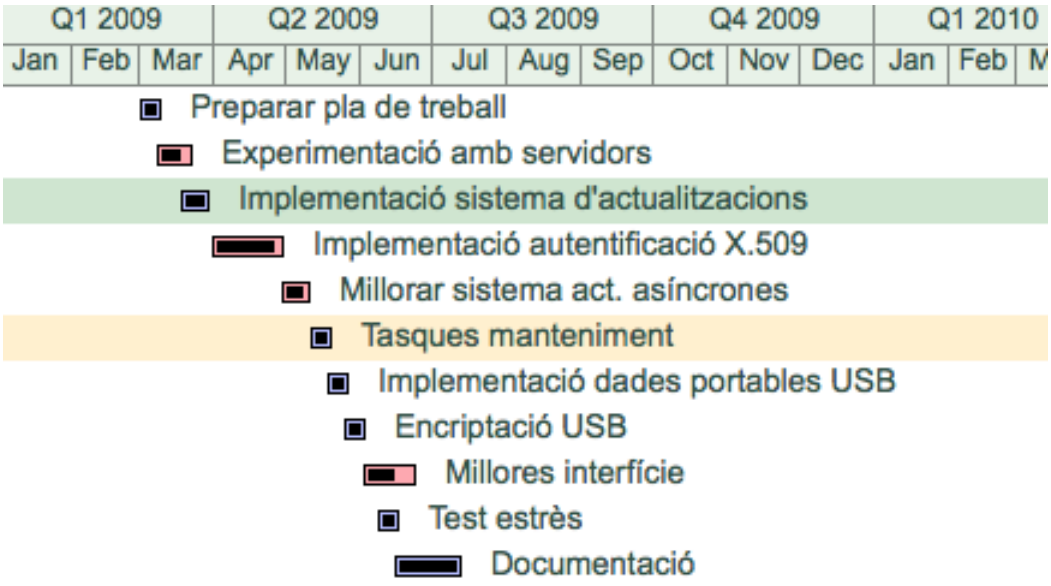


Figura 2.2: Planificació de tasques realitzades durant el projecte.

Capítol 3

Servidors XMPP

Una de les peces claus del projecte TRiEM és el servidor de missatgeria instantània. Permet unir a especialistes i pacients a través d'Internet. En l'apartat 3.1 es farà una breu descripció de les diferents alternatives dels servidors de XMPP, exposant les seves avantatges i inconvenients. A continuació, en el punt 3.2 s'exposaran els motius de la nostra elecció i el canvi que ens comporta. Finalment, en el capítol 4 es descriuran algunes millores d'usabilitat que s'han incorporat a l'aplicació.

3.1 Descripció general

És difícil fer comparacions entre aquests programes ja que es van actualitzant amb certa regularitat. Malgrat la dificultat, existeix una comparativa [17] força detallada del 2006 entre **Openfire, ejabberd i jabberd2**. Cal advertir, que alguns aspectes d'aquesta comparativa han quedat totalment obsolets (especialment la secció *Advanced*).

3.1.1 Openfire

Openfire 3.6.4 [18] (anteriorment Wildfire) és un servidor XMPP escrit en Java fet per l'empresa Jive Software. És un dels més actius amb una gran comunitat d'usuaris.

- Llicència de codi obert GNU GPL v2.
- Instal·lació i administració senzilla i amigable.
- Ofereix bones mesures de seguretat (xifratge TLS, certificats X.509).
- Extensible amb plugins (p.e. estadístiques de tràfic del servidor).
- Permet fer clustering de servidors.

Jive Software també ofereix una llibreria client en Java anomenada **Smack [19] que AXARM la fa servir** per connectar-se al servidor XMPP.

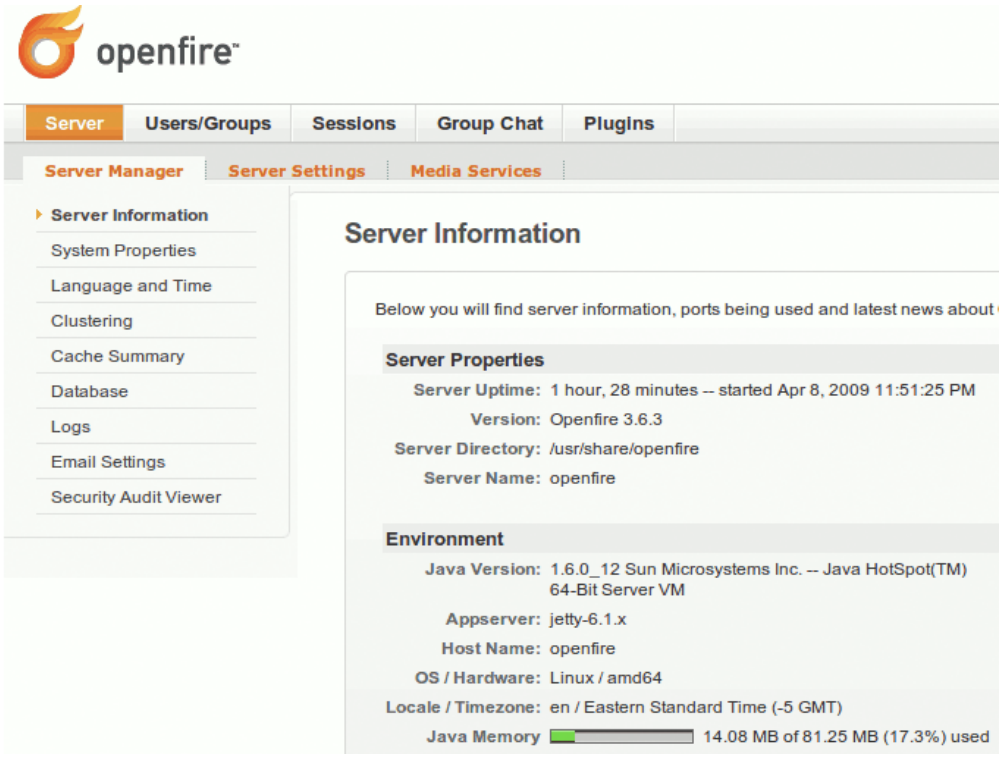


Figura 3.1: Panell de control del servidor Openfire.

3.1.2 ejabberd

ejabberd 2.0.5 [20] és una altra gran alternativa a servidors XMPP. Està escrit en Erlang per l'empresa ProcessOne, i a l'igual que Openfire disposa d'una nombrosa comunitat d'usuaris i porta un desenvolupament actiu.

Llista de característiques a destacar:

- Llicència de codi obert GNU GPL v2.
- Ofereix una bona seguretat (TLS).
- Permet aguantar milers d'usuaris alhora.
- Consumeix molt pocs recursos.
- Permet fer clustering de servidors.

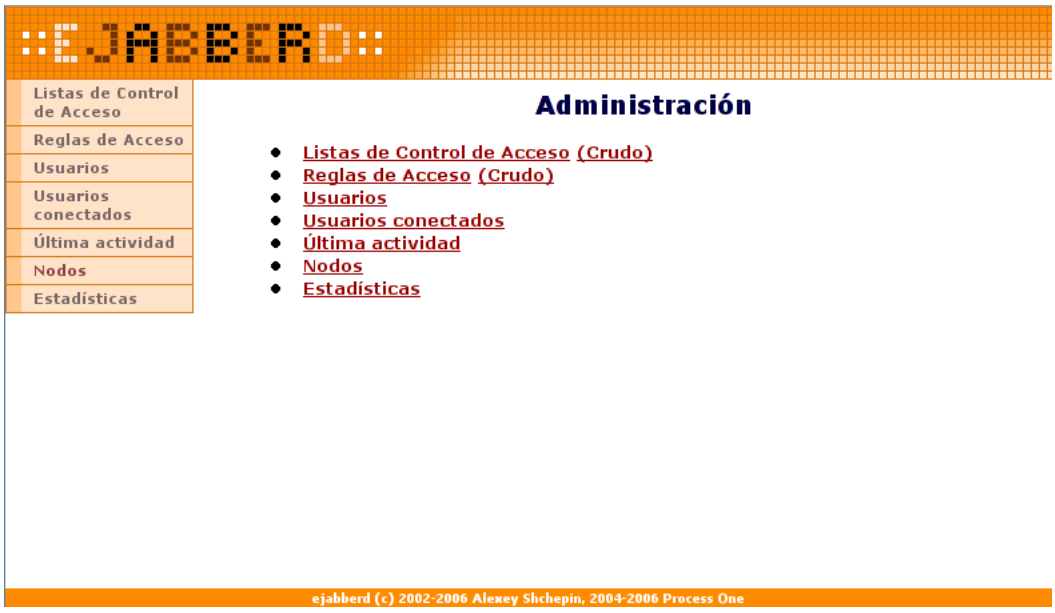


Figura 3.2: Panell de control del servidor ejabberd.

3.1.3 Jabberd2

Jabberd2 2.2.8 [21] és menys conegut que els altres dos, però no vol dir que sigui pitjor. És de codi obert GPL, encara que té un ritme d'actualitzacions inferior. Malgrat que hi ha binaris en Windows per descarregar, la versió de Linux cal compilar-la des del codi font.

3.1.4 Tigase

Tigase 4.2.0 [22] és un competidor que ha millorat recentment i a tindre en compte. També es GNU GPL v3 i ofereix moltes característiques avançades com clustering o suport de hosts virtuals. Està escrit en Java, i a l'igual que Openfire, es poden descarregar llibreries i mòduls.

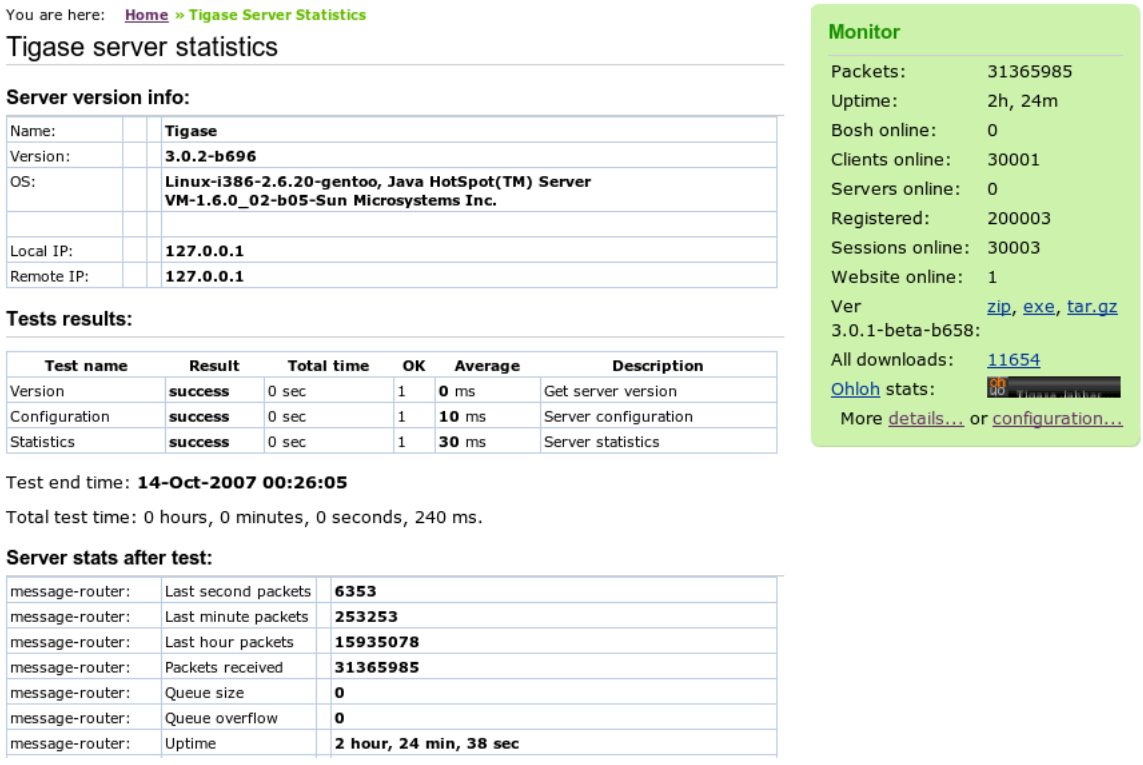


Figura 3.3: Panell de control del servidor Tigase.

3.2 Escollir un servidor XMPP pel projecte TRiEM

Des dels inicis del projecte TRiEM, ens vam decantar per fer servir ejabberd com a servidor de missatgeria instantània ja que ofereix excel·lents resultats i és àmpliament suportat. Però durant el transcurs d'aquest projecte, ens hem vist forçats a fer un canvi cap a **Openfire**.

És cert que en comparació amb Openfire, ejabberd té millor rendiment. En canvi, té dos inconvenients:

1. L'administració d'un servidor Openfire és més usable que un de ejabberd (útil per si algun dia es disposa de servei tècnic).

2. Openfire és **l'únic** servidor que suporta **autentificació a dues bandes** de certificats digitals X.509 (combinant amb Smack).

Per tant, el canvi **millora en seguretat i usabilitat** en contra d'eficiencia. Per observar quina quantitat de rendiment es perd amb el canvi, es van fer uns test de rendiment que s'expliquen en el capítol 7.

Capítol 4

Usabilitat

4.1 Millores en l'aplicació

Parlar d'usabilitat és parlar de la interacció humà-màquina. Un bon sistema és aquell que és fàcil d'entendre i de fer servir per un públic heterogeni. Per aquesta raó, és molt important de disposar de *feedback* o retroalimentació: que els propis usuaris comentin quines coses els hi costen més i alhora que necessiten per tal que es trobin còmodes.

4.1.1 Actualitzacions semi-automàtiques

Un problema força habitual quan es parla d'aplicacions d'escriptori és el de les actualitzacions. Resulta problemàtic haver d'anar a casa dels pacients cada cop que es vol actualitzar el programa (ja sigui l'aplicació en sí o una de les extensions del programa). En primer lloc, pel cost que representa el desplaçament, i en segon lloc per trobar un horari compatible amb el pacient.

El terme de semi-automàtic vol dir que no només el sistema s'ha de permetre actualitzar, sinó que cal forçar a l'usuari a fer-ho. Dit d'una altra manera, rarament sol

passar que un pacient actualitzi un component per ell sol, i més encara si l'aplicació no li notifica automàticament les actualitzacions. Per exemple, l'aplicació Firefox [23] consulta periòdicament als servidors de Mozilla la disponibilitat d'una nova versió. En cas afirmatiu, a l'usuari li apareix una nova finestra notificant-li. No només això, fins i tot alguns cops el força a fer l'actualització al reiniciar el navegador web.

La nostra idea segueix aquest model d'actualitzacions. En el panell d'estat del sistema se li notifica noves actualitzacions del programari o extensions.

4.1.1.1 Actualitzacions de les extensions

Una extensió consisteix en un sol fitxer JAR (l'equivalent a un fitxer comprimit) que conté el codi compilat i el recursos necessaris per fer-la funcionar (imatges, sons, múltiples idiomes...). El procés d'actualització és relativament senzill: només cal **substituir** el fitxer per la versió més recent i reiniciar l'aplicació.

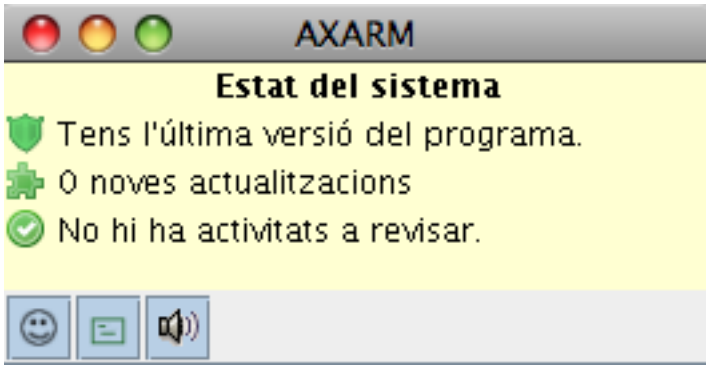


Figura 4.1: L'aplicació AXARM senyalant la part d'Estat del Sistema.

Quan l'aplicació detecta en el servidor una nova actualització d'una extensió, li apareix una nova finestra preguntant-li si vol actualitzar-ho o no. L'usuari té la possibilitat de rebutjar l'actualització, però llavors la següent vegada que engegui el programa, li tornarà a aparèixer la finestra per actualitzar. Si accepta el procés, l'aplicació descarrega totes les noves extensions i les activa.

4.1.1.2 Actualitzacions del programa principal

Realitzar una actualització de l'aplicació en marxa no és tant senzill com les extensions, ja que a l'estar en execució ens impedeix sobreescriure amb la nova versió. Adicionalment, tampoc és senzill fer que un programa es reinicii per ell sol. Per defecte, aquesta funció recau en el Sistema Operatiu i no en la màquina virtual de Java. Per tal d'afrontar aquest contratemps, es van analitzar diverses possibles alternatives:

1. **BootLoader**: El programa principal està format per una classe inicial que només es dedica a carregar extensions. La idea és que la pròpia aplicació fos com una extensió i així es resoldria el problema. És una solució elegant, però costosa a l'haver de reescriure una bona part de l'aplicació.

2. **Dos programes en memòria**: Intentar que el primer programa (original) cridi al segon (el nou) i es mori el primer sense matar el segon. Existeixen unes classes especials en Java (ProcessComunicator), però igualment és força complicat.

3. **Script**: En comptes d'executar directament el programa, l'usuari hauria d'executar un script que cridés a l'aplicació. Llavors el propi script sí podria matar i engegar la nova versió. Com a contrapartida, aquesta solució depèn de cada Sistema Operatiu i caldria disposar de tres scripts diferents (Windows, Linux, Mac OS X).

4. **Modificar en run-time**: Existeix una llibreria anomenada *Javasist* que permet modificar classes de Java en temps d'execució. No només no és elegant sinó perillós!

5. **Assistir a l'usuari**: La més simple; el programa descarrega un actualitzador en l'escriptori i se li indica a l'usuari com executar l'update. Amb dos clics s'actualitza el programa, però a canvi cal tancar i obrir el programa manualment.

Es va decidir fer la cinquena proposta ja que la seva implementació era gairebé immediata i no suposava un gran impediment en termes d'usabilitat (molts usuaris estan habituats a instal·lar aplicacions). Si a pesar de les facilitats el pacient és incapaç de fer l'actualització o si sorgeixen problemes, es poden solucionar oferint una assistència remota (es detalla en el capítol 9).

4.1.2 Calendari d'activitats

En les extensions on es creen activitats, l'especialista pot definir unes dates límit per fer-les (com si fossin deures de l'escola). Una sugerència que ens van proposar els especialistes rehabilitadors va ser la possibilitat de mostrar un calendari personalitzat:

- Especialista: Quan vol enviar una activitat, pot escollir una data de finalització simplement escollint el dia en el calendari.
- Pacient: A més de mostrar en una taula totes les activitats que té pendents, té l'opció de representar les dates en un calendari.

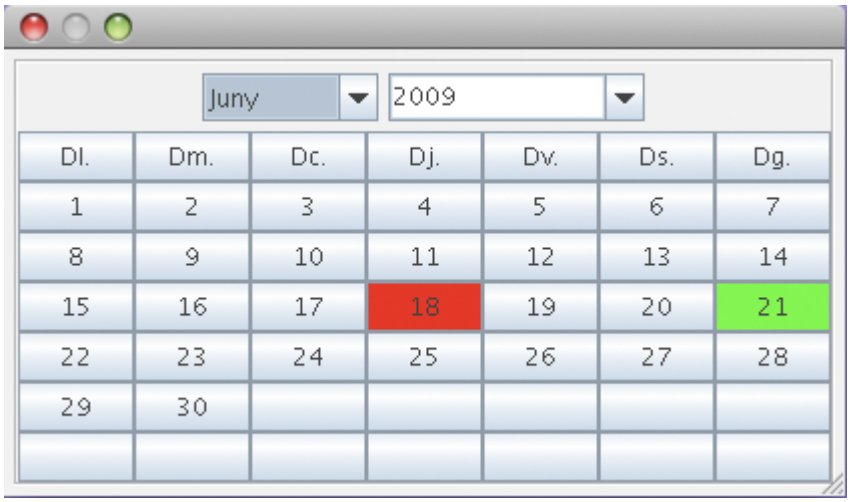


Figura 4.2: El calendari de l'especialista amb el dia actual (18) i una data triada (21).

4.2 Millores en el servei

4.2.1 Activitats asíncrones

Moltes de les propostes incorporades en l'aplicació no són més que una imitació de les tasques de la vida real. Per exemple, fer activitats síncrones de telerehabilitació és l'equivalent a fer una activitat conjunta entre l'especialista i el pacient durant una visita presencial. En canvi, al parlar d'activitats asíncrones ens referim a una nova possibilitat que no es pot fer en la vida real.

Els pacients poden compatibilitzar la seva vida laboral amb la participació en l'estudi ja que poden fer les activitats quan els hi vagi millor. Això suposa una millora en la seva qualitat de vida, al no haver de dependre d'horaris concrets. A més a més, el sistema obté una **retroalimentació** o *feedback* que li permet realitzar un seguiment personalitzat del pacient (no podrà dir allò de que "ja he fet totes les activitats"). El pacient també rep retroalimentació perquè sap si està fent bé o no les activitats.

Durant el transcurs d'aquest projecte de màster s'han **creat vàries activitats** que funcionen en mode síncron i asíncron. Cada una d'elles disposa de simples variacions disponibles com oferir diferents nivells de dificultat o fer-la més amena. En acabar qualsevol activitat, els resultats (temps consumit, encerts i errors, gràfics, ajuda demanada, ...) queden enregistrats i a disposició dels especialistes.

En totes les activitats desenvolupades fins ara (excepte la de la catifa de ball) és necessari que el pacient disposi d'una palanca de comandament o *joystick*, perquè es treballa a nivell oculo-motor, la coordinació bimanual i unimanual (fina i gruixuda), flexió i extensió del canell, força, manipulació i precisió, així com l'atenció, la concentració, rapidesa visuperceptiva, percepció i discriminació visual.

4.2.1.1 Extensió memòria

L'activitat consisteix en **memoritzar** un conjunt d'elements que apareixen a la pantalla durant un temps limitat. Després s'han de reconèixer i identificar amb el *joystick* aquests elements dins d'un conjunt més gran d'elements similars (de la mateixa "família").

Variacions de l'extensió:

- Graduar el temps donat per a memoritzar els elements.
- Graduar la dificultat amb el nombre d'elements a memoritzar i el nombre d'elements que es mostren per pantalla.
- Graduar el temps de realització de l'exercici.

Les tres variacions es representen amb un conjunt de nivells (fins a 7 nivells diferents).

Figura 4.3: Activitat de la memòria.

4.2.1.2 Extensió carretera

A la pantalla surt una carretera i amb el joystick s'ha de **resseguir el recorregut** correcte fins arribar a la meta. El circuit sempre comença des de la cantonada superior esquerra i finalitza a la cantonada inferior dreta. Si el cotxe surt de la carretera, ha de tornar al principi del circuit.

Variacions de l'extensió:

- El traçat de la carretera determina la dificultat (més estreta, més girs...).
- També es pot dissenyar la carretera com un laberint incorporant carreteres falses per tal de posar a prova les facetes cognitives.

Dissenyar les carreteres és molt simple: només cal fer servir un editor d'imatges i dibuixar la traça negra sobre un fons blanc.

Figura 4.4: Activitat de la carretera.

4.2.1.3 Extensió macedònia

En l'extensió de la macedònia la pantalla es divideix en dues parts per una línia vertical. A cada part de la pantalla surten varis elements que van caient. Quan hi hagi dos elements que siguin iguals, llavors s'han de seleccionar amb el joystick abans que arribin al final de la pantalla.

Variacions de l'extensió:

- Els elements són imatges que poden representar nombres, paraules, lletres, ...
- Es pot graduar la velocitat en que cauen els elements (per nivells).
- L'activitat pot acabar en un temps donat, en realitzar l'exercici un cert nombre de vegades o en aconseguir un nombre d'encerts predeterminat.
- El pacient rep respostes visuals i sonores de si ho està fent bé o malament.

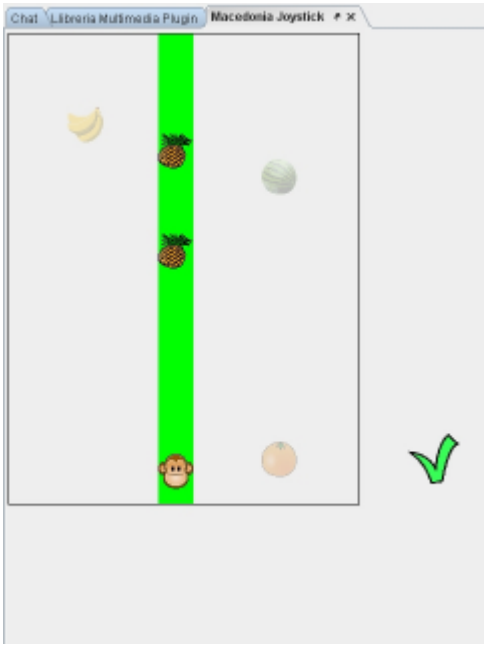


Figura 4.5: Activitat de la macedònia.

4.2.1.4 Extensió catifa de ball

Les anteriors activitats fan treballar la mà (és a dir, les extremitats superiors). Amb la incorporació d'una catifa de ball, es poden realitzar activitats **d'extremitats inferiors**. L'especialista estableix una seqüència de moviments a fer (peus, mans o genolls) i el nombre de repeticions. El pacient pot fer l'exercici malament perquè el programa només pot **detectar si es fa contacte a la catifa** o no, i per tant podria fer trampa. És recomanable fer aquesta activitat amb la supervisió d'un especialista.

Variacions de l'extensió:

- Es pot utilitzar una seqüència predeterminada o entrar manualment una nova seqüència.

Figura 4.6: Activitat utilitzant una catifa de ball.

4.2.1.5 Extensió figures

Semblant a l'activitat de la memòria, al pacient se li presenta una figura geomètrica durant uns segons i després l'ha de reproduir amb el joystick.

Variacions de l'extensió:

- Es pot memoritzar la figura i després fer-la, o simplement que es mostri per pantalla i l'hagi d'imitar.
- Graduar el temps de memorització i el temps de realització de l'exercici.

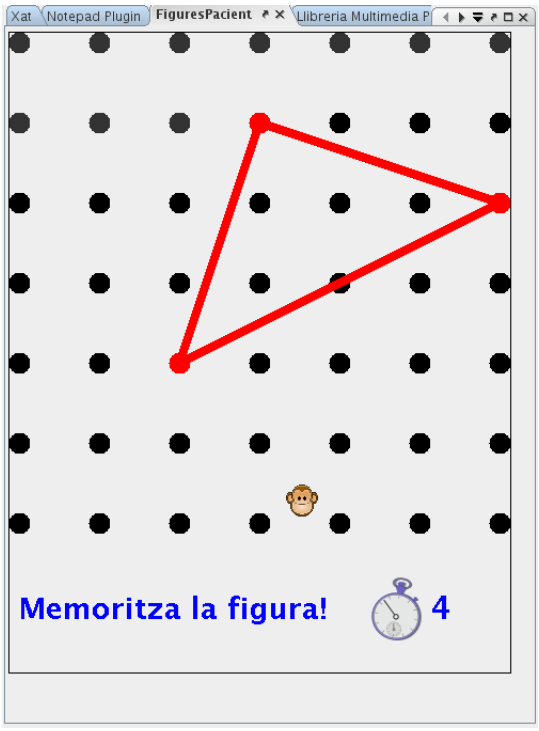


Figura 4.7: Activitat on el pacient ha de memoritzar una figura geomètrica per reproduir-la a continuació.

Capítol 5

Seguretat: Autenticació

Un dels punts que més s'han treballat en aquest projecte és el de la seguretat. Es pot subdividir en tres aspectes claus: el xifratge TLS/SSL de les comunicacions, l'autenticació d'usuaris via certificats digitals i el xifratge local de les dades dels usuaris. En primer lloc, s'explicarà un breu resum del que són aquests conceptes per tot seguit explicar amb detall el que s'ha realitzat.

5.1 Conceptes clau

5.1.1 Certificats X.509 i entitats certificadores CA

X.509 és un estàndard fet servir per implementar un sistema de claus públiques (PKI) perquè defineix com ha de ser un certificat i com validar-lo. Es basa en un sistema d'entitats certificadores (CA) **jerarquitzat**. A diferència d'altres models com OpenPGP (*Pretty Good Privacity*), només les CA són capaces de generar certificats vàlids. X.509 també inclou un sistema de llistes de revocació (CRL) de certificats caducats o invàlids.

Una entitat certificadora (CA) és una empresa que emeteix certificats digitals pel

seu ús de tercers. La confiança dels usuaris amb la CA és clau, ja que depèn exclusivament d'aquests determinar si una CA és fiable o no. Les grans CA (com Verisign) disposen d'un gran reconeixement mundial i es classifiquen com entitats certificadores de primer nivell. Molts navegadors web moderns incorporen les seves claus per defecte, sota pagament previ. No només les grans companyies són CA, també els estats, institucions i governs disposen de les seves pròpies CA, generalment per ús només dins la seva jurisdicció.

A Espanya, la *Fábrica Nacional de Moneda y Timbre* [24] és l'encarregada de crear certificats digitals a nivell nacional (és una CA). Una utilitat d'aquest certificat és poder fer la declaració de la renda per Internet. El nou DNI-E (certificat fet per la policia, no per la FNMT) incorpora un microxip que es pot fer servir amb un lector de tarjetes. A Catalunya, la Generalitat de Catalunya disposa de l'Agència Catalana de Certificació [25], una entitat pública que emet certificats per administracions públiques catalanes. Les entitats de registre (com els ajuntaments) són les encarregades de crear els certificats pels ciutadans i/o les empreses.

Figura 5.1: Imatge del nou DNI Electrònic extreta de la web www.dnielectronico.es

5.1.2 Xifratge TLS/SSL

TLS (Transport Layer Security, sucessor de SSL - Secure Sockets Layer -) és un protocol criptogràfic que ofereix connexions segures a través d'Internet realitzant criptografia sobre les dades. Per establir una connexió segura es realitzen aquests tres passos:

1. Negociar entre les dues parts quins algorismes es faran servir.
2. Intercanvi de claus públiques i autentificació (a una o dues bandes).
3. Xifrat simètric del tràfic.

Del punt 1, algorismes de clau pública comuns són RSA o DSA, mentre que xifratges actuals són Triple DES o AES. Del punt 2, l'autenticació a una banda significa que només es valida el servidor, mentre que a dues el client està obligat a identificar-se correctament (amb certificats digitals).

En AXARM fent servir el servidor Openfire, s'ha **implementat una autenticació X.509 a dues bandes**. La UdG disposa d'una entitat certificadora i genera certificats digitals als especialistes i pacients. Per fer la CA i tots els certificats s'ha escollit el software **OpenSSL** [26] degut a que és de codi lliure i ha demostrat una gran robustesa.

En l'apèndix A s'explica tot el procés per crear-les.

5.1.3 Java Keystore

A partir de les últimes versions del Java, s'incorpora un nou sistema per treballar amb els certificats digitals i *keystores*. Un *keystore* és un fitxer codificat en binari on es guarden totes les claus i entitats certificadores a les que pot accedir el Java. Les claus solen ser privades i es guarden amb una protecció extra (una contrasenya), mentre que les CA que hi ha són amb les que confiem.

El gran avantatge de fer servir aquest tipus de magatzem de claus és que **no** depèn del Sistema Operatiu (com per exemple el clauer del sistema en Mac OS X).

Per treballar amb aquest tipus de fitxers, disposem de la utilitat **keytool** que es crida des de la línia de comandes (ve amb el JRE de sèrie).

Listing 5.1: Llistar claus privades o CA

```
keytool -list -keystore "nom_fitxer" -v
```

Listing 5.2: Afegir-ne

```
keytool -import -keystore "nom_fitxer" -alias "nom_unic"
-file "fitxer.pem"
```

Listing 5.3: Eliminar-ne

```
keytool -delete -keystore "nom_fitxer" -alias "nom_unic"
```

5.2 Per què fer servir X.509?

Per fer-se'n una idea de què és un certificat, se'l pot considerar com un *token* o objecte personal. Realitzant un símil amb la realitat, el certificat seria equivalent a la tarja de crèdit, o a les claus del cotxe. Sense ell no pot funcionar el programa.

Si el certificat es fa servir com a possessió, es podria pensar en fer servir altres sistemes com les *cookies*. Però hi ha una sèrie d'avantatges que disposa el X.509 sobre les cookies:

- Està estandaritzat: assegura compatibilitat amb moltes altres aplicacions.
- No és manipulable ja que sinó el certificat es detecta com invàlid.
- Permet guardar i llegir fàcilment altres dades (nom, correu, adreça...) que amb una *cookie* seria molt complicat.

5.3 Integrar certificats en el servidor Openfire

Un cop situats en el context, explicarem com crear la infraestructura necessària per utilitzar autentificació a dues bandes amb X.509 en el nostre servidor XMPP. Per aconseguir-ho, m'ha estat de gran ajuda els **forums d'Openfire** [27], ja que en la documentació oficial no s'explica com fer aquest procés, i gràcies als autors d'Openfire he pogut resoldre tots els meus dubtes.

Abans de començar, cal haver generat les claus pel servidor i, al menys, per a un client.

1. Cal fer servir l'última versió d'Openfire (3.6.4 en aquests moments).
2. Accedir al WebAdmin via navegador web (http://localhost:9090).

3. Afegir les següents propietats en Server Manager → System Properties.

Taula 5.1: Paràmetres del servidor

sasl.mechs	EXTERNAL,PLAIN,DIGEST-MD5
xmpp.client.cert.policy	needed
xmpp.client.certificate.accept-selfsign	false
xmpp.client.certificate.cert	/ruta/absoluta/.der
xmpp.client.certificate.verify	true
xmpp.client.certificate.verify.chain	true
xmpp.client.certificate.verify.root	true
xmpp.client.certificate.validity	true

4. Reemplaçar el certificat del servidor pel nostre en Server Settings → Server Certificates (Encara que digui "Pending Verification"funciona correctament).

   Nota: Aquest pas equival a afegir el certificat en el keystore del servidor.

5. Anar a Server Settings → Security Settings i marcar **Required** en Client Connection Security. També es **desactiva** el login anònim en Server Settings → Registration & Login

6. A continuació, cal afegir la nostra CA en el keystore del servidor manualment:

Listing 5.4: Afegir la CA creada en client.truststore

```
cd $openfire/resources/security
keytool -import -keystore client.truststore
-alias "url_servidor" -file "cacert.pem"
//Contrasenya per defecte: "changeit"
```

   Per comprovar que s'ha afegit correctament, si fem un *list* apareixerà una **trustedCertEntry**.

7. Per seguretat, es pot repetir el mateix pas però amb el fitxer truststore.

5.4 Integrar certificats en els clients

Per cada client s'ha de crear un **nou fitxer keystore** que contingui tant el certificat del client com la seva clau privada (és necessària la parella). Un detall important és que els clients **no necessiten la CA**. Amb la utilitat keytool es pot generar un nou keystore però per fer-ho correctament cal anar amb compte quan es vol afegir un certificat, més clau privada alhora.

Segons la documentació existent, *keytool no permet importar una clau privada existent per la qual ja disposem del seu certificat importat prèviament.* Seguint les recomanacions de la web d'AgentBob [28], cal passar abans el nostre certificat client i la seva clau privada del format PEM a DER, i llavors fer servir un petit programa propi anomenat **ImportKey** per afegir-les (el codi font del programa es troba en la web citada anteriorment).

Listing 5.5: Crear keystore del client

```
cd ./.jbother/profiles/default
//PEM a DER
openssl pkcs8 -topk8 -nocrypt -in client-priv.pem -inform PEM
-out key.der -outform DER
openssl x509 -in client-cert.pem -inform PEM -out cert.der
-outform DER

/*Editar parametres Importkey.java: keypass, defaultalias
i keystorename*/
javac Importkey.java
java Importkey key.der cert.der
```

Per comprovar que s'ha afegit correctament, si fem un *list* apareixerà una **keyEntry**.

5.5 Implementar l'ús de PKI en l'aplicació

Per acabar la secció sobre certificats digitals, explicarem quins canvis s'han afegit en el codi font per utilitzar aquesta nova funcionalitat. La llibreria **Smack 3.1.0** [19] s'encarrega d'accedir al keystore per nosaltres, i de facilitar el certificat amb clau privada al servidor com s'observa en el següent codi.

Nota: A data d'aquest escrit, és necessari **aplicar una actualització no oficial** a Smack (o descarregar la versió del Subversion).

```
import org.jivesoftware.smack.*;
import org.jivesoftware.smackx.*;

ConnectionConfiguration cc = new ConnectionConfiguration
(server, port, server);
cc.setKeystorePath(JBother.profileDir + File.separatorChar + "keystore");
cc.setKeystoreType("jks");
cc.setCallbackHandler(new CallbackListener());
XMPPConnection connection = new XMPPConnection(cc);
//System.setProperty("javax.net.debug", "ALL");
```

Un CallbackHandler és una classe especial de Java *(javax.security.auth.callback)* pensada per satisfer les peticions d'autenticació que li indica el servidor. Segons com s'implementa, li pot preguntar o no a l'usuari per les dades. Exemples de **Callbacks** són NameCallback (demana nom d'usuari) o PasswordCallback (contrasenya).

Per tal que funcioni, cal implementar un ***PasswordCallback*** com a continuació:

```
import javax.security.auth.callback.Callback;
import javax.security.auth.callback.NameCallback;
import javax.security.auth.callback.PasswordCallback;

public class CallbackListener implements CallbackHandler{

public CallbackListener() {
}
public void handle(Callback[] callbacks) throws IOException,
    UnsupportedCallbackException {

    for (int i = 0; i < callbacks.length; i++) {
        if (callbacks[i] instanceof PasswordCallback) {
               //Keystore
       callback.setPassword("changeit".toCharArray());
               //També pot demanar password de auth
        }
        else if (callbacks[i] instanceof NameCallback) {
                NameCallback name = (NameCallback) callbacks[i];
                name.setName(ConnectorThread.getInstance().getUsername());
        }
        else {
                throw new UnsupportedCallbackException
                (callbacks[i], "Unrecognized Callback");
        }
    }
}
```

Ja només ens queda fer la connexió i l'autenticació al servidor XMPP.

```
SASLAuthentication.supportSASLMechanism("DIGEST-MD5", 2);
SASLAuthentication.supportSASLMechanism("PLAIN", 1);
SASLAuthentication.supportSASLMechanism("EXTERNAL", 0);
try{
   connection.login(username, null, resource);
}
catch (XMPPException e) {
   if (e.getXMPPError() != null) {
     errorMessage = resources.getString("xmppError"
        + e.getXMPPError().getCode());
   }
}
```

L'autentificació externa significa que el servidor no demanarà contrasenya ja que s'ha verificat el certificat del client. Si no està disponible, es prova amb el clàssic usuari/contrasenya.

5.6 Exemple d'ús

La propietat **javax.net.debug** posada a *ALL* ens serveix per *debugar* i veure tot el tràfic que circula a l'hora de fer el handshake entre client i servidor. El següent exemple segueix l'ordre d'execució del client:

Listing 5.6: Exemple de tràfic SSL I: Keystore local

```
KeystoreCallback
***
found key for : songohan.udg.edu
chain [0] = [
[
  Version: V3
  Subject: CN=test@songohan.udg.edu, ...
  Signature Algorithm: SHA1withRSA, OID = 1.2.840
  Key:  Sun RSA public key, 4096 bits
  modulus: ...
  public exponent: 65537
  Validity: [From: Mon May 18 17:15:57 CEST 2009,
              To: Thu May 16 17:15:57 CEST 2019]
  Issuer: CN=songohan.udg.edu, ...
  SerialNumber: [    03]
]]
***
```

En primer lloc, el keystore local ha trobat una clau privada que concorda amb la direcció del servidor que ens connectem. Si el servidor li pregunta per un certificat, li donarà aquest.

Listing 5.7: Exemple de tràfic SSL II: Intercanvi de certificats

```
*** ClientHello, TLSv1
RandomCookie: GMT: 1245666783 bytes
Session ID: {}
Cipher Suites: ...
*** ServerHello, TLSv1
RandomCookie: GMT: 1245667101 bytes
Session ID: {74, 63, 95, ...}
Cipher Suite: SSL_RSA_WITH_RC4_128_MD5
*** Certificate chain
chain [0] = [[
  Version: V1
  Subject: CN=songohan.udg.edu, ...
  Signature Algorithm: SHA1withRSA, OID = 1.2.840
  Key: Sun RSA public key, 4096 bits
  modulus: ...
  public exponent: 65537
  Validity: [From: Mon May 18 16:56:15 CEST 2009,
              To: Thu May 16 16:56:15 CEST 2019]
  Issuer:CN=songohan.udg.edu, ...
  SerialNumber: [    d80d12a8 f9db299f]
]]
*** CertificateRequest
Cert Types: RSA, DSS,
Cert Authorities:
<EMAILADDRESS=contacte@triem.org, CN=songohan.udg.edu...>
*** ServerHelloDone
```

A continuació es posen d'acord amb el protocol a utilitzar. El servidor comença enviant el seu certificat al client, i alhora li **demana** el certificat del client.

Listing 5.8: Exemple de tràfic SSL III: Autenticació

```
matching alias: songohan.udg.edu
*** Certificate chain ...

*** ClientKeyExchange, RSA PreMasterSecret, TLSv1
Random Secret: { 3, 1, 218, 224, 227...}

SESSION KEYGEN
PreMaster Secret ...

CONNECTION KEYGEN
Master Secret ...

*** CertificateVerify
*** Finished
verify_data: { 87, 116, ...}
***
```

En l'últim pas, el client observa que disposa d'un certificat apte pel nom del servidor, i li envia aquest al servidor. En l'etapa *CertificateVerify* el servidor comprova si el certificat és vàlid, i en cas afirmatiu li permet continuar al client.

Per acabar, si ens interessa accedir manualment al keystore i obtenir els certificats que hi ha, es pot fer servir les classes ***java.security.Keystore*** i ***java.security.cert.**;*** ens pot resultar útil per obtenir més dades com el correu electrònic, la província...

```
//Obrim el fitxer keystore
FileInputStream fis = new FileInputStream(JBother.profileDir
+ File.separatorChar + "keystore");
KeyStore key = KeyStore.getInstance("jks");
key.load(fis,"changeit".toCharArray());

//escollim el certificat que concorda amb el servidor XMPP
Certificate cert = key.getCertificate(server);
if (cert instanceof X509Certificate) {
     X509Certificate x509Cert = (X509Certificate) cert;
     String str = x509Cert.getSubjectX500Principal().getName();
}
```

Capítol 6

Seguretat: Confidencialitat

Un cop vista la part del xifratge de les comunicacions i de l'autenticació dels usuaris, queda per resoldre el tema de la seguretat de les dades. Durant una sessió normal de l'aplicació, es van generant continguts com són les converses, resultats d'activitats, fotos o vídeos de les webcams. Totes aquestes dades s'han de tractar de manera sensible perquè són dades mèdiques.

La llei aplicable a dades mèdiques és la LOPD (Ley Orgnánica de Protección de Datos) [29], que disposa d'un apartat especial per aquest cas. Les dades relatives a la salut tenen la consideració de dades especialment protegides ja que formen part de la intimitat i privacitat de l'individu. Malgrat que aquestes es consideren d'alt nivell segons la LOPD, moltes de les dades generades per l'aplicació no tenen la gravetat que tindrien d'altres com el fitxer de malalts del VIH o el fitxer de donants.

Per aquesta raó, i per tal de complir amb el nivell més exigent de la LOPD, és **obligatori** que totes les dades que s'emmagatzemen siguin xifrades (a l'igual que les comunicacions). Si un pacient desitja disposar de més seguretat, se li ofereix una bona solució que s'explicarà a continuació.

6.1 Dades en dispositius extraïbles

Un dels canvis importants que s'ha realitzat a l'aplicació és l'ús de memòries extraïbles (tant targetes de memòria SD com llapis de memòria (*pendrives*) o altres formats físics). Hi ha una sèrie d'avantatges per utilitzar-les:

- És bo separar les dades de l'ordinador perquè en cas de robatori les dades no es veurien compromeses.
- Alhora, en el cas que hi hagués algun problema amb l'aplicació o l'ordinador, la recuperació de les dades seria immediata.
- Utilitzar una memòria extraïble equivaldria a tenir d'una tarjeta de crèdit o les claus del cotxe, encara que no és el mateix (si et canvien el cotxe, la clau no et val).

Cada una de les memòries dels pacients està personalitzada amb el seu certificat digital. **L'aplicació sense aquestes dades personals no es pot engegar**, augmentant així la seva seguretat.

Implementació: l'aplicació segueix el següent esquema quan s'engega:

1. Mira en el $HOME de l'usuari el fitxer *axarm.dat*. Si existeix, intenta trobar les dades en la ruta que li indiqui.
2. Si no les troba, llavors detecta en quin S.O. s'està executant i decideix:
    - Windows: Obté les unitats de disc disponibles i fa una cerca.
    - Linux: Mira en la carpeta /media (on Ubuntu monta els pendrives)
    - Mac OS X: Mira en la carpeta /Volumes
3. Si no troba els fitxers, avisa a l'usuari que introdueixi la seva memòria i que torni a provar.

4. Mentre no trobi els fitxers, l'aplicació no començarà.

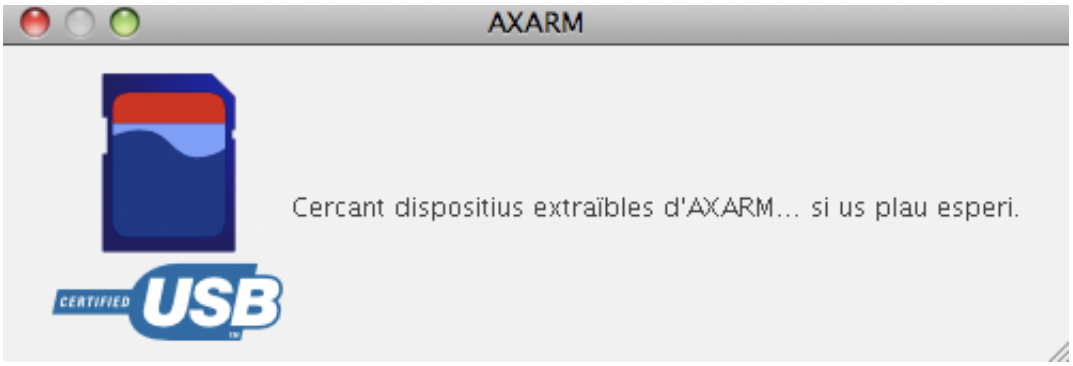


Figura 6.1: Mostrant a l'usuari que estem cercant dispositius extraïbles.

El rendiment de l'aplicació és força bo quan escriu les dades a la memòria ja que només s'observa un lleuger retard quan el programa fa l'accés al dispositiu. A més a més, els preus d'aquests dispositius són molt **econòmics** i cada cop en fan de més capacitat.

Per exemple, MediaMarkt oferta al juliol de 2009 una memòria USB de 8 GB per 10,99€. IVA inclòs.

Tots aquests canvis segueixen la filosofia del TRiEM, que un dels objectius bàsics és poder oferir el servei a **baix cost**.

6.2 Xifratge

6.2.1 Dades dels pacients amb TrueCrypt

El següent pas, després de traslladar totes les dades a la memòria, és la de realitzar el xifratge. A mode de breu explicació, **xifrar** dades significa codificar-les per tal que siguin il·legibles per una tercera persona que no sàpiga interpretar-les.

Per poder accedir a les dades xifrades, cal conèixer una contrasenya. Per tant, si l'usuari vol més seguretat llavors disposarà de menys usabilitat ja que s'haurà d'enrecordar d'un pas extra a l'hora d'utilitzar l'aplicació.

Un requeriment del xifratge és que ha de funcionar en els tres principals S.O (Windows, Linux i Mac), i alhora no ha de dependre d'AXARM. La solució ens l'ofereix un programa anomenat **TrueCrypt** [30].

- Gratuït i de codi obert.
- Permet xifrar tant fitxers com unitats de disc o dispositius sencers.
- El xifratge és automàtic, en temps real i transparent per l'usuari.
- Algorismes de xifratge disponibles: AES-256, Serpent i TwoFish.

Figura 6.2: Logo del programa TrueCrypt.

Per crear fitxers xifrats es segueix un assistent del TrueCrypt. La guia d'usuari [31] detalla molt bé aquest procés, a l'igual que la documentació en la seva pàgina web.

Passos per xifrar les dades:

1. Instal·lar TrueCrypt i executar-lo.

2. Clic en: Create Volume.

3. Tipus de volum: Seguirem el recomenat que és File Container (un sol fitxer).

4. Demana si volem un volum estàndard o ocult. Un volum ocult significa que dins del mateix fitxer hi pot haver dues unitats xifrades diferents. Per posar un exemple, en una disposem de dades que treballem habitualment i a l'altra, de dades molt més sensibles. En cas que ens forcin a revelar la contrasenya, li podem dir la *falsa* i d'aquesta manera enganyar-los. En contra, hi ha la possibilitat de que les dades dels dos volums es barregin i es perdin, sense que es pugui saber si això passarà. Escollirem **estàndard**.

5. Nom del fitxer: Un bon consell per augmentar la seguretat és intentar dissimular el volum xifrat el màxim possible. A aquesta tècnica se la coneix com **esteganografia**, que és l'art d'ocultar missatges dins d'altres anomenats portadors. Compte, que l'esteganografia no vol dir criptografia (l'art de xifrar dades).

6. Xifratge: Escollim l'algorisme *AES* perquè ofereix molt bona relació velocitat/-xifratge conjuntament amb el *hash* SHA-512.

7. Espai: És important ja que un cop creat no es pot modificar. En un memòria de 2 GB, el fitxer hauria d'ocupar un 75% i deixar un 25% lliure per altres fitxers normals.

8. Contrasenya: Molt important, un altre cop surt el tema **seguretat vs usabilitat**. TrueCrypt recomana una contrasenya mínima de 20 caràcters: l'ideal seria una contrasenya llarga però fàcil d'enrecordar-se'n.

9. Finalment, moure el ratolí uns 30 segons per generar uns bons nombres aleatoris i esperar a que acabi. Formatejar 1,5 GB sol trigar mitja hora.

Dins del mateix TrueCrypt podem muntar i desmuntar la unitat xifrada que acabem de crear. Cal anar en compte de no esborrar el fitxer que hem creat, ja que llavors s'eliminaria tota la unitat xifrada.

6.2.2 Xifratge del servidor

Fins ara s'ha detallat l'ús del xifratge en els entorns dels clients, però també es pot aplicar el xifratge en el servidor. Al treballar amb dades mèdiques cal assegurar-se'n la seva integritat en la banda dels clients i del servidor. L'aspecte positiu més destacable és que ofereix una capa extra de protecció, en l'hipotètic cas que un atacant aconsegueixi entrar en el sistema (ja sigui físicament o remotament). En contra, cal posar una contrasenya cada cop que es reinicia el servidor.

En entorns Linux es disposa de diverses opcions a l'hora de xifrar:

- Xifrar una partició en concret, o tot el disc dur.
- Fer servir un xifratge a baix nivell del propi Sistema Operatiu, o utilitzar el mateix TrueCrypt.

L'elecció d'un o altre sistema dependrà de les nostres necessitats en quan velocitat/nivell de xifratge.

6.3 Automuntatge de les dades xifrades

Per tal de facilitar el procediment del muntatge d'unitats xifrades, es va pensar en la idea de fer servir sistemes d'automuntatge. És a dir, que al connectar la memòria externa automàticament es munti el volum xifrat preguntant per la contrasenya.

Encara que resulta **molt pràctic** (i a més a més, aporta més usabilitat), debilita un pèl la seguretat ja que se li donen pistes a un possible atacant. A més a més, cal destacar que el procediment de muntatge varia segons el sistema operatiu, i no es fa de la mateixa forma en Windows, Linux o Mac.

- **Windows:** TrueCrypt disposa d'una eina que prepara automàticament una memòria extraïble per utilitzar la funció de **Reproducció Automàtica** si està activada.

  Cal anar al menú *Tools* → *Traveler Disk Setup* i seguir aquests passos:

  - Trobar a quina unitat es troba connectat el pendrive.
  - Desmarcar: *Include TrueCrypt Volume Creation Wizard.*
  - Autorun: Marcar *Auto-mount TrueCrypt volume.* Escollir el fitxer del volum xifrat i una unitat lliure (ex: Z:).

  En realitat, aquesta utilitat ens crea un fitxer autorun.ini preparat amb les nostres necessitats.

Listing 6.1: Contingut del fitxer autorun.ini.

```
[autorun]
label=TrueCrypt Traveler Disk
icon=TrueCrypt\TrueCrypt.exe
action=Mount TrueCrypt volume
open=TrueCrypt\TrueCrypt.exe /q background /lZ /e /m
rm /v "Terminator␣Salvation␣(V.O.)␣2009␣DVDRip.avi"
shell\start=Start TrueCrypt Background Task
shell\start\command=TrueCrypt\TrueCrypt.exe
shell\dismount=Dismount all TrueCrypt volumes
shell\dismount\command=TrueCrypt\TrueCrypt.exe /q /d
```

Quan connectem la memòria ens apareix la següent imatge, la qual permet muntar la unitat xifrada si escollim la primera opció.

Figura 6.3: Diàleg que surt al posar el pendrive en un Windows.

Addicionalment, per **entorns Windows no és necessari tindre el programa TrueCrypt instal·lat en l'ordinador ja que s'inclou en el dispositiu.**

- **Linux (Ubuntu):** És necessari haver instal·lat prèviament el TrueCrypt en l'ordinador de l'usuari per tal de poder muntar la unitat xifrada. A l'igual que en Windows, s'ha creat un petit shell script anomenat *autorun* que s'autoexecuta al connectar-lo a l'ordinador. El nostre script està preparat per entorns gràfics KDE, GNOME i en mode consola.

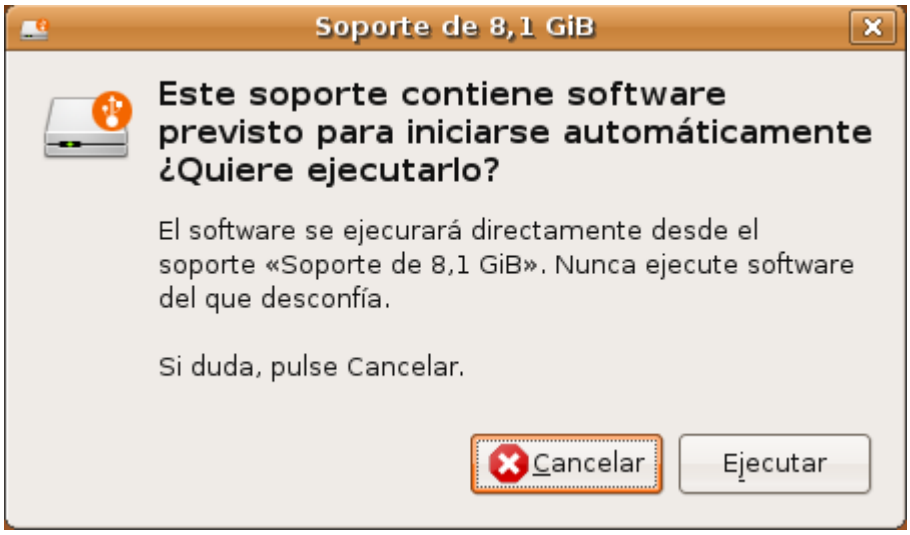


Figura 6.4: Diàleg que surt al posar el pendrive en un Ubuntu.

- **Mac OS X:** És necessari haver instal·lat prèviament el TrueCrypt en l'ordinador de l'usuari per tal de poder muntar la unitat xifrada. Per polítiques de seguretat del Sistema Operatiu, no es pot fer cap script que s'autoexecuti. Malgrat això, s'han fet dos scripts: un per muntar la unitat i l'altre per desmuntar-la.

Listing 6.2: AppleScript per muntar unitats xifrades.

```
tell application "Finder"
set currFolder to (folder of the front window as string)
end tell

set p to POSIX path of currFolder
set fileName to quoted form of
"Terminator␣Salvation␣(V.O.)␣2009␣DVDRip.avi"

do shell script
"/Applications/TrueCrypt.app/Contents/MacOS/TrueCrypt␣"
 & quoted form of p & fileName & "␣/Volumes/AXARM"
```

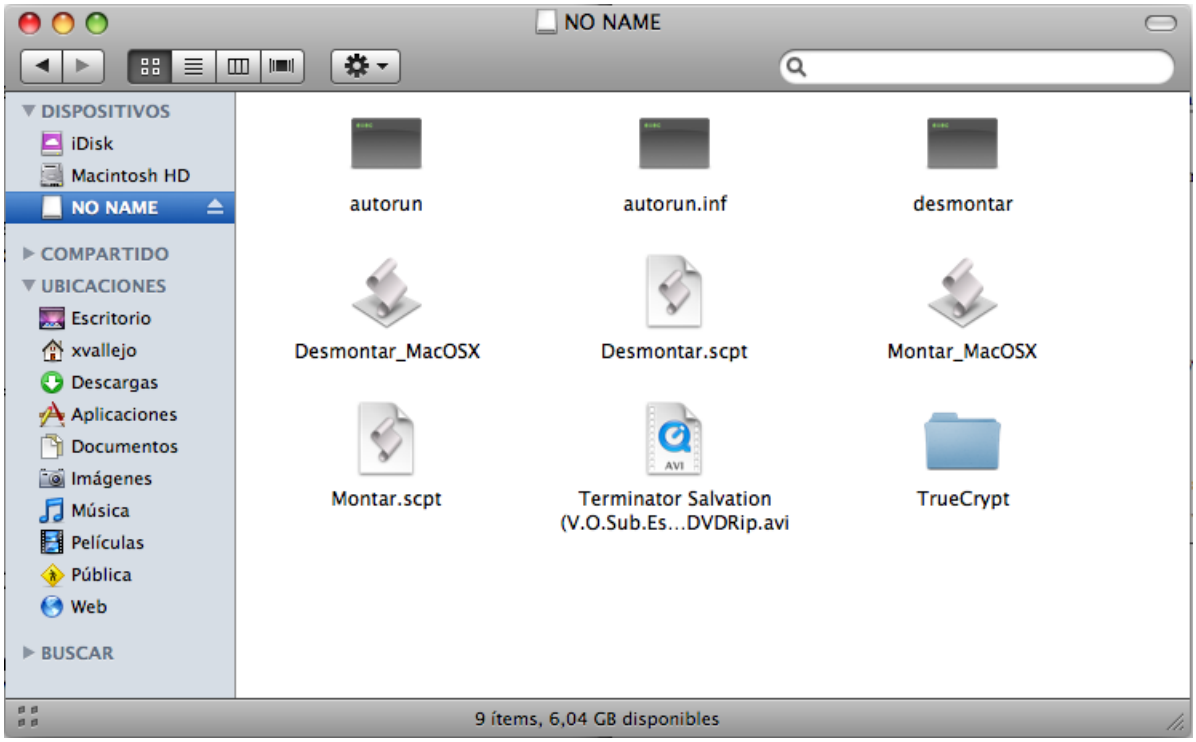


Figura 6.5: Estructura de la memòria preparada pels tres sistemes operatius.

Capítol 7

Escalabilitat

7.1 Tsung: programa per estressar servidors

Tsung [32] és una eina de codi obert (GNU GPL v2) per provar càrregues distribuïdes multi-protocol. Serveix per posar a prova tant servidors HTTP, MySQL com XMPP. El seu propòsit és **simular múltiples usuaris** per testejar l'escabilitat i rendiment d'aplicacions IP basades en client/servidor. El mateix paquet genera una pàgina HTML amb els resultats en forma estadística i gràfica.

L'estratègia consisteix en distribuir el programa en unes quantes màquines client i llavors cadascuna pot simular centenars o milers d'usuaris concurrentment. Tots els usuaris simulats segueixen unes instruccions que s'indiquen en un fitxer de configuració XML.

Tsung està creat per la mateixa empresa que ejabberd [20], i també està implementat en el llenguatge Erlang.

7.2 Experiments reals amb servidors XMPP

7.2.1 Preparació prèvia

El primer pas és **instal·lar-lo**: des de la pàgina web ens descarreguem el paquet per l'Ubuntu. Si volem que generi gràfics cal instal·lar també el programa **gnuplot**. Amb la comanda *sudo apt-get install gnuplot* és suficient.

Per poder fer els tests, primer cal **crear els usuaris ficticis** en el servidor XMPP. Per l'estudi clínic que vol realitzar la FEM, com a molt tenen previst 60 usuaris no simultanis. Com que pensem que 60 són pocs usuaris, les proves que realitzarem són amb 1000 usuaris, dels quals es connectaran el màxim nombre que aguanti el servidor.

Com que introduir a mà 1000 usuaris és una feina molt lenta i tediosa, cal crear un script que automatitzi el procés. El procés varia segons el servidor XMPP triat:

- **ejabberd**: Crear el següent shell script en la carpeta del binari ejabberdctl (canviant *servidor* per l'adreça real) i **executar-lo**.

Listing 7.1: Afegir usuaris amb ejabberd

```
#!/bin/sh
for i in `seq 1 1000`;
do
echo $i && ./ejabberdctl register tsung$i
servidor p4ssw0rd$i
done
```

Un cop realitzats els tests, per **eliminar els usuaris ficticis** es pot fer servir el mateix script però canviant register per unregister, i sense el camp password.

- **OpenFire**: El procediment és un pèl més complicat però igualment satisfactori:

  1. Des del WebAdmin, instal·lar el plugin oficial **User Service**.
  2. Anar a *Server Settings* → *User Service* i activar-ho.
  3. Crear un script en php i executar-ho via web.

Listing 7.2: Afegir usuaris amb openfire

```
<?php
for ($i = 1; $i <= 1000; $i++){
$f = fopen("http://url.servidor:9090/plugins/
userService/userservice?type=add&secret=contrasenya
&username=tsung".$i."&password=p4ssw0rd".$i,"r");

$response = fread($f,1024);

if (ereg('ok', $response)){
        echo "Usuari $i afegit correctament!";
        echo "<br/>";
}
else{
        echo "Error en afegir l'usuari $i";
        echo "<br/>";
}
fclose($f);
}
?>
```

  4. És convenient desactivar opcions de monitorització, com guardar converses, i sortir del WebAdmin.

Per **eliminar** els usuaris, es pot reaprofitar el mateix script però canviant add per delete i sense el camp &password.

Un detall important és que per poder fer bé la prova, cal **desactivar el xifratge SSL i la petició de certificats** en el servidor XMPP.

7.2.2 Contingut del fitxer de configuració

Degut a que escriure tot el contingut del fitxer XML trencaria l'estructura del document, el podreu trobar sencer en l'apèndix B.

Fent un breu comentari del XML, en l'apartat <load> s'especifiquen les fases del test.
S'han creat tres fases ben diferenciades:

- Fase 1: Durant 1 minut s'anirà connectant un usuari cada 0,01 segons.
- Fase 2: Durant 5 minuts s'aniran connectant a un ritme de 0,1 segons.
- Fase 3: Durant 3 minuts s'anirà connectant un usuari cada segon.

En total el test dura 10 minuts, i la càrrega va de més intensitat a menys. S'ha limitat a un màxim de 1000 usuaris a la vegada i tots es generaran a partir d'una sola màquina.

També es defineix en el XML el que fa cada usuari, que es detalla a continuació:

1. Connecta al servidor.

2. Intenta autentificar-se i espera als altres clients que també s'autentifiquin.

3. Es posa visible i obté la seva llista de contactes.

4. Envia un missatge curt a un contacte online.

5. Afegeix un contacte nou a la seva llista de contactes.

6. Envia dos missatges a dos contactes.

7. Canvia el nom del perfil.

8. Envia un missatge a un usuari offline.

9. Elimina un contacte de la seva llista.

10. Desconnecta.

Com que envia missatges a usuaris online i offline, això farà que provoqui errors al servidor. Finalment, per executar el test:

- El fitxer cal posar-lo en *$HOME/.tsung/tsung.xml*
- Escriure en una consola de comandes: ***tsung start***

Un cop executat el test obtindrem els *logs* del test. Amb un script en PERL del Tsung crearem les pàgines HTML amb els resultats.

1. Anar al directori on ha generat els resultats: $HOME/.tsung/log/Carpeta_Resultats
2. Executar: ***/usr/lib/tsung/bin/tsung_ stats.pl***

És possible que doni un error el PERL del tipus: "Can't locate Template.pm in INC". Per arreglar-ho, cal instalar el paquet Template del PERL desde la mateixa consola del sistema:

```
> sudo su
> perl -MCPAN -eshell

// Demana per fer una configuracio manual: respondre NO
> install Template

// Automaticament es descarrega i ens va preguntar
// anar contestant amb ENTER (per defecte)
```

7.2.3 Resultats

S'han realitzat les proves en dues màquines diferents: una de lenta i una altra de més ràpida. Totes dues disposen de les últimes versions d'ejabberd i OpenFire, amb el qual s'han fet 4 tests en total.

Màquina 1: Intel Celeron 2,6 Ghz, 2GB RAM i Ubuntu 8.04

Màquina 2: Intel Core 2 Duo E4500 a 2,2 Ghz, 2 GB RAM i Ubuntu 9.04

Tsung dóna resultats en forma de **mitjana cada 10 segons**, com per exemple el temps de resposta del servidor o el nombre de transaccions per segon. D'aquesta manera el programa de proves disposa de més temps per obtenir les dades.

7.2.3.1 Màquina 1: OpenFire

A continuació es mostra un resum dels resultats numèrics i gràfics de les proves:

Estadístiques principals:

Nom	↑ 10s mitjana	↓ 10s mitjana	Ràtio més alt	Mitjana	Nº
connect	0.420 s	1.33 ms	29.4 / s	14.76 ms	3759
page	3.540 s	0.358 ms	89.1 / s	0.180 s	28434
request	3.231 s	1.51 ms	34 / s	0.284 s	7268
session	37.7 s	10.1 s	28.8 / s	21.1 s	3759

Estadístiques transaccions:

Nom	↑ 10s mitj.	↓ 10s mitj.	Ràtio més alt	Mitjana	Nº
tr_authenticate	6.471 s	3.75 ms	20.2 / s	0.770 s	3674
tr_close	0.788 s	0.200 ms	25.6 / s	36.66 ms	3471
tr_offline	0.873 s	2.27 ms	22.9 / s	77.71 ms	3471
tr_online	1.033 s	1.25 ms	58.7 / s	0.105 s	10477
tr_roster	2.659 s	1.07 ms	19.3 / s	0.273 s	3582
tr_rosteradd	1.706 s	0.305 ms	20.5 / s	0.159 s	3516
tr_rosterdelete	1.038 s	0.294 ms	24.8 / s	60.99 ms	3471
tr_rosterrename	1.143 s	0.838 ms	23.3 / s	97.55 ms	3471

Errors:

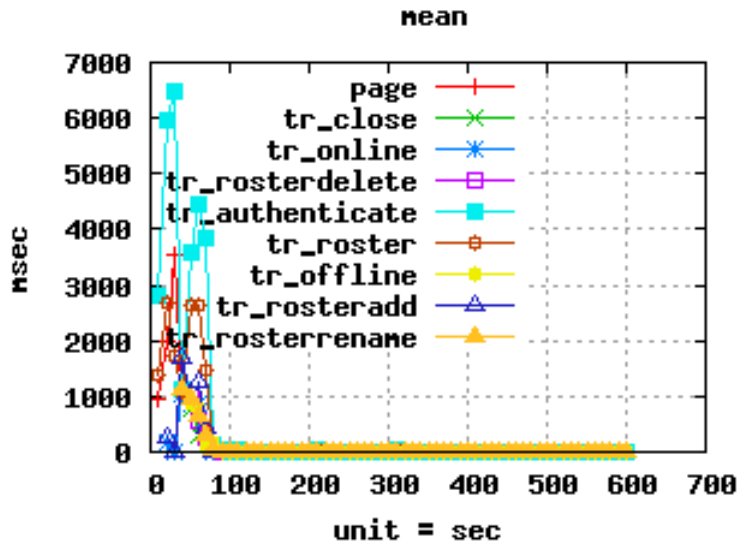
Nom	Ràtio més alt	Nombre total
error_unknown	28.8 / s	288

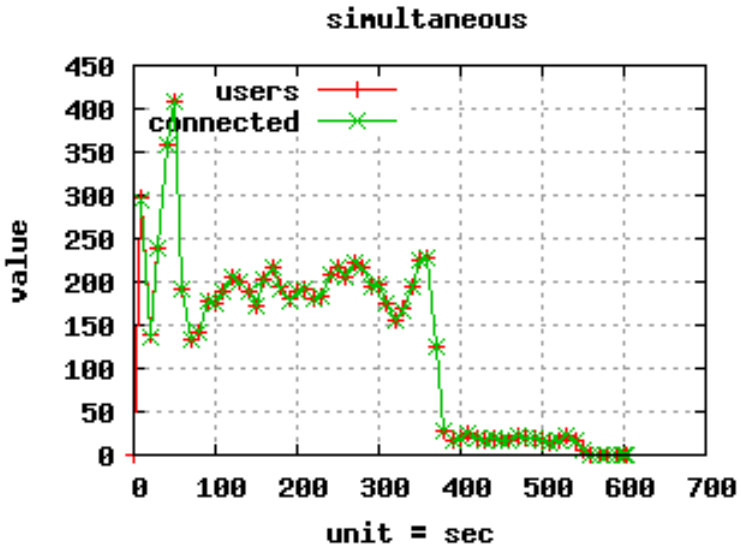
Màxim nombre d'usuaris connectats a la vegada: 408

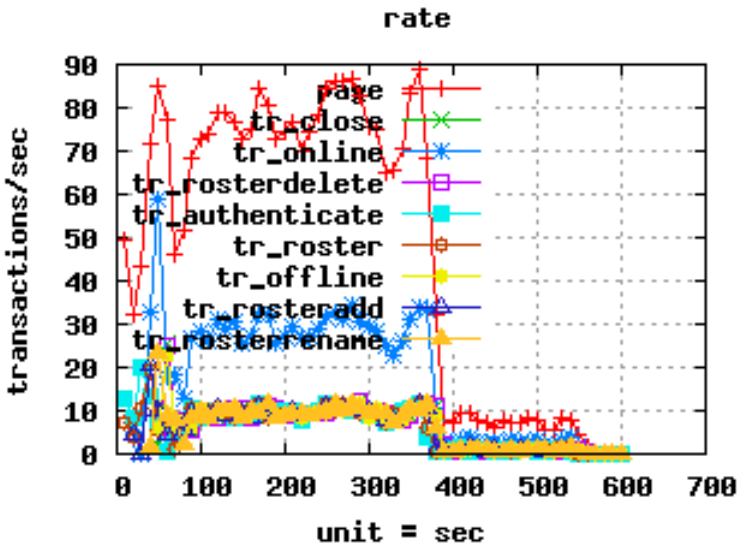
Comentari: La primera gràfica representa el temps de resposta del servidor, la segona el nombre simultani d'usuaris i la tercera el nombre d'operacions per segon.

Observant les dades numèriques i la gràfica dels temps de resposta, en els primers 60 segons (fase 1) el servidor rep moltes peticions d'autentificació i li costa molt poder-les acceptar ja que hi ha casos en que la mitjana per autenticar-se arriba als 7 segons d'espera.

En canvi, el nombre d'errors de la primera fase és acceptable però el nombre d'operacions per segons és força baix ja que no arriba ni a les cent operacions per segon.







7.2.3.2 Màquina 1: ejabberd

Estadístiques principals:

Nom	↑ 10s mitjana	↓ 10s mitjana	Ràtio més alt	Mitjana	Nº
connect	5.082 s	0.626 ms	82.4 / s	0.753 s	5416
page	0.549 s	0.275 ms	259.2 / s	0.117 s	41659
request	0.284 s	0.855 ms	93.9 / s	64.39 ms	10659
session	25.3 s	3.741 s	47.1 / s	20.0 s	5416

Estadístiques transaccions:

Nom	↑ 10s mitj.	↓ 10s mitj.	Ràtio més alt	Mitjana	Nº
tr_authenticate	58.69 ms	1.89 ms	61.5 / s	22.79 ms	5415
tr_close	1.94 ms	0.185 ms	45.3 / s	0.676 ms	5117
tr_offline	1.81 ms	0.317 ms	70 / s	0.806 ms	5117
tr_online	3.22 ms	0.303 ms	120.5 / s	0.971 ms	15351
tr_roster	1.143 s	1.06 ms	51.6 / s	0.124 s	5243
tr_rosteradd	10.13 ms	0.289 ms	53.2 / s	1.58 ms	5117
tr_rosterdelete	1.87 ms	0.267 ms	45.3 / s	0.765 ms	5117
tr_rosterrename	2.33 ms	0.216 ms	50.9 / s	0.808 ms	5117

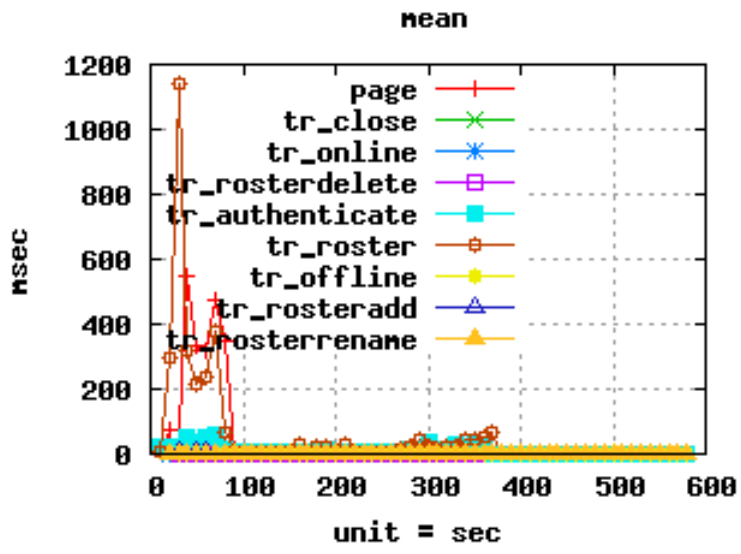
Errors:

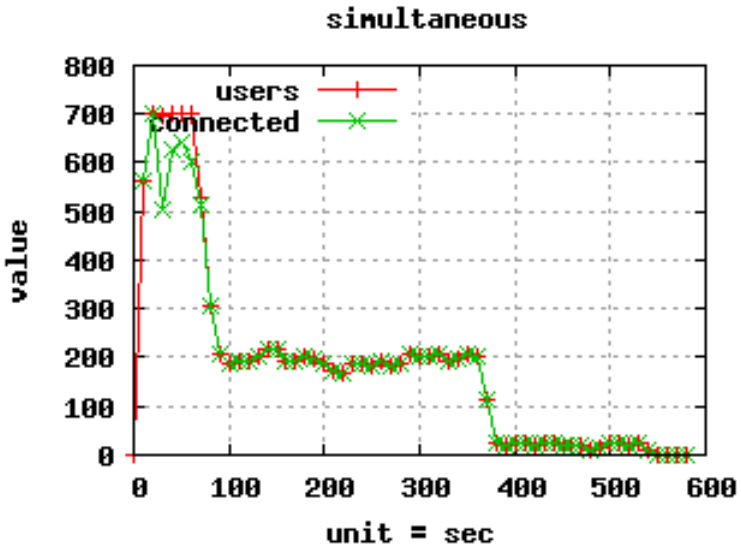
Nom	Ràtio més alt	Nombre total
error_no_free_userid	73.4 / s	2879
error_no_offline	39.9 / s	735
error_unknown	29.9 / s	299

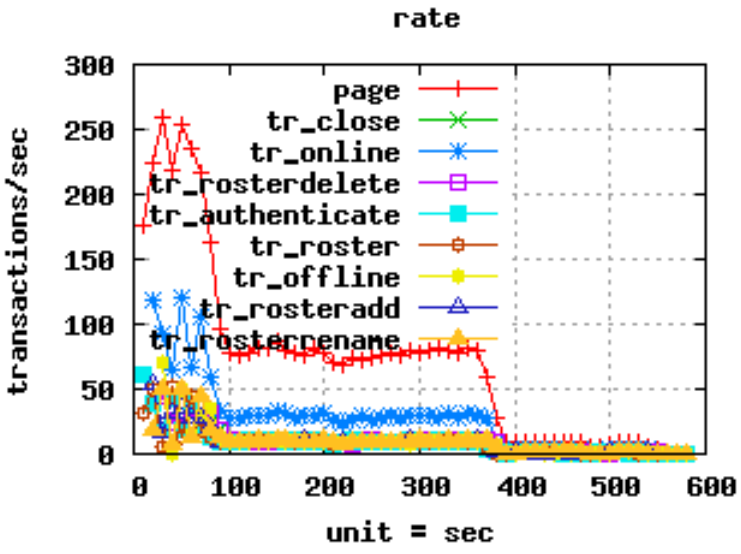
Màxim nombre d'usuaris connectats a la vegada: 701

Comentari: A diferència d'OpenFire, ejabberd es comporta molt millor en la resposta a les operacions dels clients, tant en temps com en nombre d'operacions per segon. A més a més, també pot suportar més clients alhora que el seu competidor, però el nombre d'errors és un pèl superior respecte a Openfire.

Un cop finalitzades les proves amb aquesta màquina, es pot afirmar que amb **recursos limitats, ejabberd obté més rendiment que Openfire**.







7.2.3.3 Màquina 2: Openfire

Estadístiques principals:

Nom	↑ 10s mitjana	↓ 10s mitjana	Ràtio més alt	Mitjana	Nº
connect	5.18 ms	2.40 ms	79.6 / s	3.71 ms	5520
page	9.02 ms	0.766 ms	299 / s	2.40 ms	42483
request	6.38 ms	0.541 ms	99.9 / s	2.44 ms	10863
session	20.1 s	3.719 s	50.8 / s	19.2 s	5520

Estadístiques transaccions:

Nom	↑ 10s mitj.	↓ 10s mitj.	Ràtio més alt	Mitjana	Nº
tr_authenticate	17.48 ms	1.64 ms	64.3 / s	6.90 ms	5519
tr_close	5.12 ms	0.181 ms	49.6 / s	1.08 ms	5220
tr_offline	3.26 ms	0.345 ms	70 / s	1.32 ms	5220
tr_online	2.90 ms	0.345 ms	125.7 / s	1.16 ms	15660
tr_roster	6.91 ms	0.462 ms	58.4 / s	2.54 ms	5344
tr_rosteradd	2.42 ms	0.276 ms	57.9 / s	0.915 ms	5220
tr_rosterdelete	4.37 ms	0.387 ms	49.6 / s	1.61 ms	5220
tr_rosterrename	4.08 ms	0.238 ms	58.4 / s	1.11 ms	5220

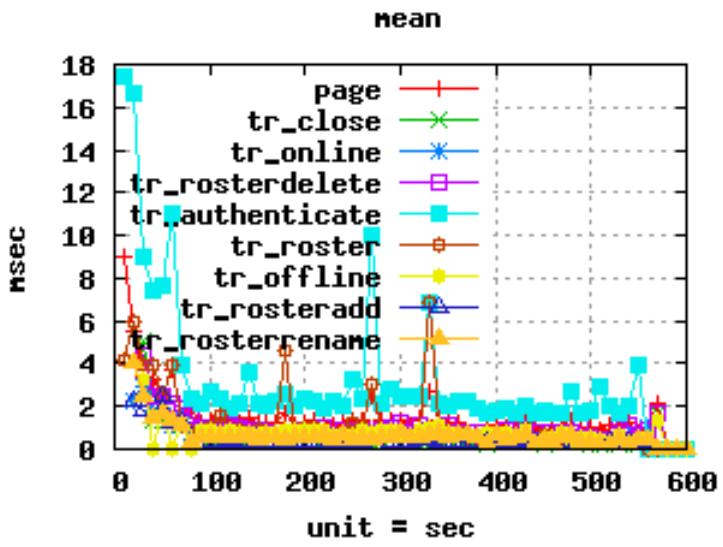
Errors:

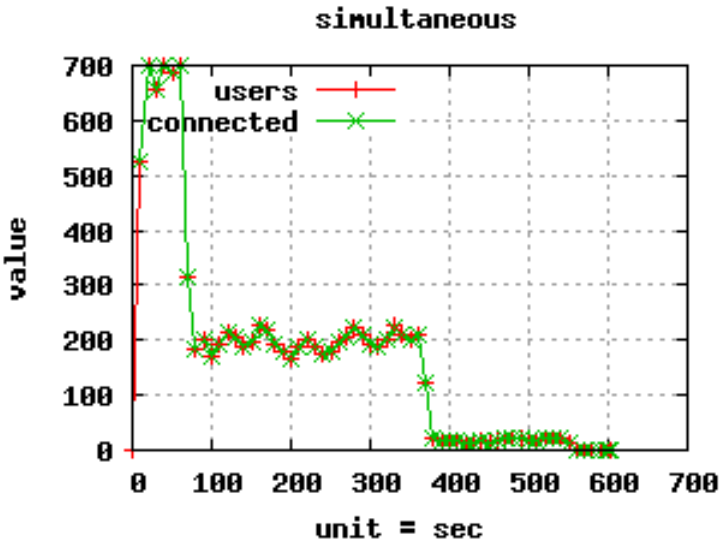
Nom	Ràtio més alt	Nombre total
error_no_free_userid	64.2 / s	2783
error_no_offline	32.8 / s	832
error_unknown	30 / s	300

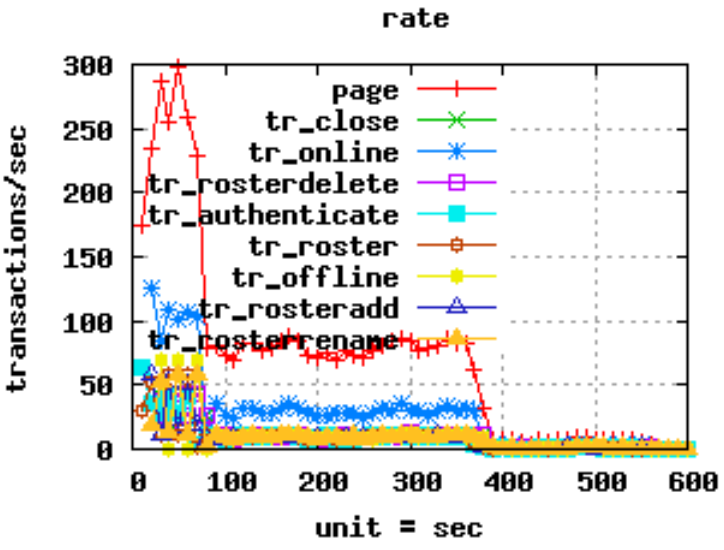
Màxim nombre d'usuaris connectats a la vegada: 700

Comentari: Al disposar d'una màquina amb més recursos, s'observa que el temps de resposta a peticions ha baixat fins als ms, a l'igual que ha augmentat el nombre d'usuaris connectats simultàniament i el nombre d'operacions per segon.

Igualment, s'observen similituds amb l'anterior test d'Openfire ja que l'operació que triga més en realitzar el servidor és l'autenticació.







7.2.3.4 Màquina 2: ejabberd

Estadístiques principals:

Nom	↑ 10s mitjana	↓ 10s mitjana	Ràtio més alt	Mitjana	Nº
connect	6.23 ms	2.10 ms	86.6 / s	3.35 ms	5475
page	4.08 ms	0.189 ms	304.6 / s	1.19 ms	42114
request	5.04 ms	0.546 ms	99.2 / s	1.47 ms	10764
session	20.0 s	3.666 s	59.9 / s	19.1 s	5475

Estadístiques transaccions:

Nom	↑ 10s mitj.	↓ 10s mitj.	Ràtio més alt	Mitjana	Nº
tr_authenticate	10.77 ms	1.15 ms	67.7 / s	2.37 ms	5474
tr_close	0.447 ms	0.142 ms	56.9 / s	0.249 ms	5175
tr_offline	0.506 ms	0.208 ms	70 / s	0.346 ms	5175
tr_online	0.791 ms	0.215 ms	141.6 / s	0.389 ms	15525
tr_roster	8.89 ms	0.682 ms	52.7 / s	2.12 ms	5290
tr_rosteradd	0.687 ms	0.258 ms	64.5 / s	0.414 ms	5175
tr_rosterdelete	0.561 ms	0.182 ms	56.9 / s	0.323 ms	5175
tr_rosterrename	0.671 ms	0.156 ms	52.5 / s	0.322 ms	5175

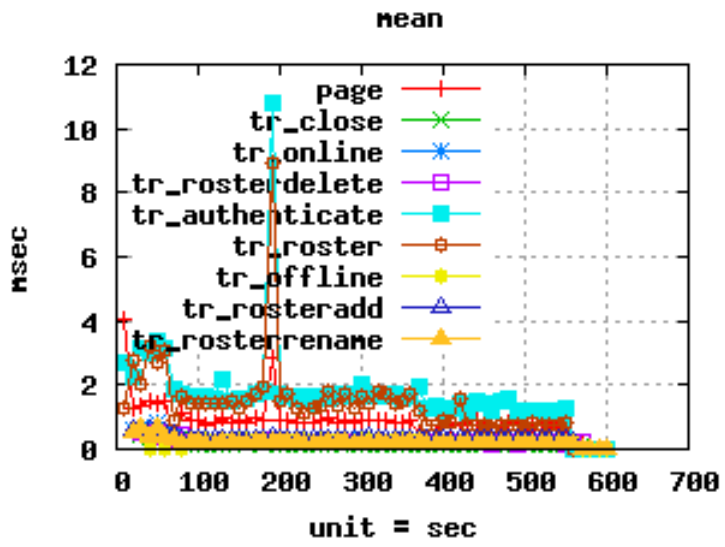
Errors:

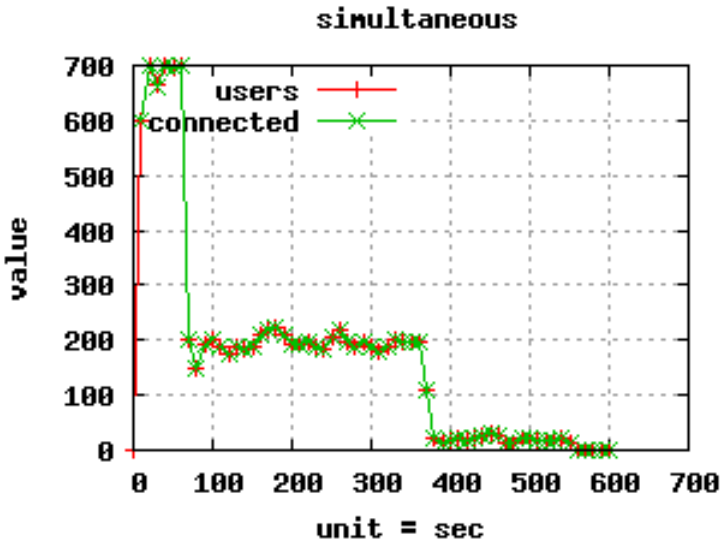
Nom	Ràtio més alt	Nombre total
error_no_free_userid	73.5 / s	2911
error_no_offline	28.4 / s	807
error_unknown	30 / s	300

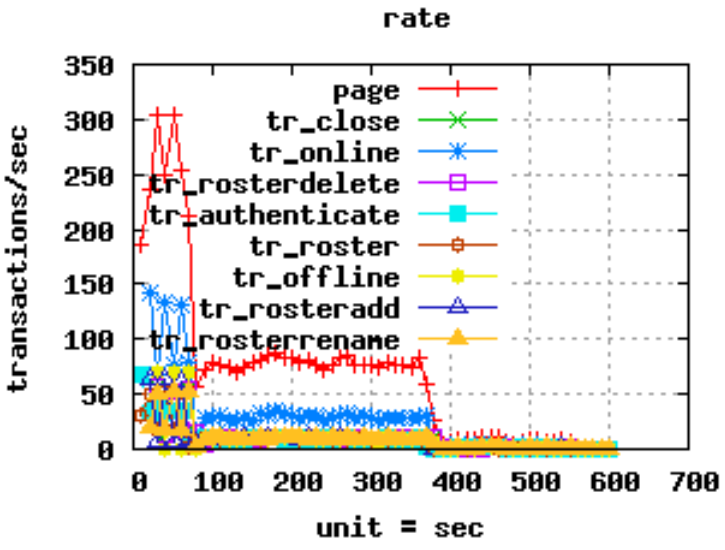
Màxim nombre d'usuaris connectats a la vegada: 700

Comentari: El rendiment en aquesta prova és excel·lent (a l'igual que amb Openfire), i no hi ha molt més a comentar.

Està clar que amb una màquina com un Core2Duo, li caldria aplicar una prova molt més agressiva per veure resultats (com per exemple, muntar un clúster de servidors XMPP). Però si recordem **l'objectiu inicial**, que era el de suportar uns seixanta usuaris connectats a la vegada, llavors es conclueix que **l'escabilitat del sistema està resolta**.







Capítol 8

SDK per crear una extensió

En aquest capítol explicarem els detalls tècnics per tal de poder crear una nova extensió sense haver de conèixer tota l'aplicació.

8.1 Estructura d'una extensió

Una extensió o *plugin* només pot ser **un sol fitxer JAR** (en el fons, no és res més que un fitxer comprimit que conté el codi a executar i altres recursos com imatges o sons). En l'arrel de l'extensió ha d'existir un fitxer anomenat ***plugin.properties*** que conté els següents camps:

8.1.1 Propietats

mainClass: "com.valhalla.jbother.plugins.Nom_Classe_Plugin", és la classe principal que s'intentarà executar quan es carregui un *plugin*.

description: Una breu descripció.

name: Nom de l'extensió.

APIVersion: Versió comuna als *plugins* (91214).

version: La versió que utilitzem per comprovar si hi ha noves actualitzacions.

author: Autor de l'extensió.

releaseDate: Un data com per exemple: Jun 12, 2008.

os: Sistema operatiu (si es vol per tots els SO es posa "all").

arch: Si el plugin està dirigit per una arquitectura concreta o no ("all").

Els dos últims paràmetres es comproven juntament amb les variables que es troben en execució en la màquina de Java.

Els següents paràmetres són **opcionals** i afecten a la visibilitat del *plugin* dins de la finestra del xat:

leftSide: Si té el valor true, la pestanya del *plugin* es situarà a la part esquerra, en cas contrari a la dreta.

hide: Si té el valor true, el *plugin* no tindrà cap pestanya en la finestra (no es veurà), però seguirà funcionant. És útil per fer extensions que són invisibles a l'usuari.

twoPanels: Si té el valor true, l'aplicació intentarà carregar dues vegades el *plugin* per mostrar-lo en dues finestres (una a dalt i l'altre a sota, com en la videoconferència).

Utilitzar-lo implica fer servir els següents atributs:

- **Name1:** El nom de la pestanya de dalt.
- **Name2:** El nom de la pestanya de sota.

El codi font s'organitza dins d'un package del Java: *com.valhalla.jbother.plugins*. La classe principal (mainClass) ha d'implementar tota la interfície del *plugin* i tots els *plugins* han d'heretar els mètodes i implementar-los.

8.1.2 Mètodes obligatoris

- **public boolean init()** Mètode de la classe principal de l'extensió que es crida quan l'aplicació la carrega. El mètode s'encarrega d'inicialitzar-la i registrar-la pels diferents events que pugui rebre. Un petit fragment de codi de com seria un mètode init:

```
import com.valhalla.jbother.JBother;
import com.valhalla.pluginmanager.*;
public boolean init()
{
PluginChain.addListener (this);
//Inicialitzar variables, mètodes...
com.vahalla.Logger.debug("Plugin initiated");
return true;
}
```

- **public void unload()** Representa l'acció contrària del mètode anterior, quan es descarrega el *plugin* del programa.

```
import com.valhalla.jbother.JBother;
import com.valhalla.pluginmanager.*;
public void unload()
{
PluginChain.removeListener( this );
}
```

  El *plugin* s'ha de treure de la cadena de plugins en aquest mètode, per tal que es puguin carregar i descarregar dinàmicament en el programa principal.

- **public Object crear(Object obj)** Aquest mètode **crea l'objecte** i permet la **comunicació entre el programa principal i el plugin**.

L'aplicació principal crida aquest mètode, i l'objecte de sortida espera que sigui un JPanel per mostrar-lo per pantalla. El *plugin* s'encarrega de crear l'objecte sense que intervingui l'aplicació. Dit d'una altra forma, el programa principal no sap realment què és el que s'executa, només sap que el que rebrà, ho mostrarà per pantalla.

8.1.3 Events

Tots els events que el programa principal crea els poden rebre les extensions. Si una extensió vol actuar al rebre un determinat event, llavors cal implementar el següent codi en el *plugin*:

```
public void handleEvent (PluginEvent event)
{
if (event instanceof ConnectEvent) {
   //fer el codi necessari per l'event
}
else {
   //altres events...
}
```

Existeixen varis events ja programats, com quan es connecta al servidor, quan surt del programa... També se'n poden crear de nous si és necessari.

8.2 Panell d'opcions propi

Si es vol disposar d'un panell d'opcions propi en les opcions del programa, llavors cal afegir més comandes en els codis init() i unload().

```
import com.valhalla.jbother.JBother;
import com.valhalla.pluginmanager.*;
import com.valhalla.jbother.preferences.PreferencesDialog;
public boolean init()
{
PluginChain.addListener(this);
PreferencesDialog.registerPluginPanel("Nom",new JPanel());
return true;
}
```

La nova línia de codi (PreferencesDialog) serveix per afegir un nou JPanel al menú de les opcions. Per crear-lo, ens cal declarar una nova classe en un altre package diferent dins el JAR de l'extensió: ***com.valhalla.jbother.preferences***.

Aquesta classe nova derivada ha d'implementar els següents mètodes:

- getSettings()
- getPreferencesPanelName() Ha de retornar el nom del Plugin, i és important que el nom contingui la paraula "Plugin".

A més a més, si es vol guardar dades en un fitxer de text, cal implementar aquests dos mètodes:

- setSettings()
- writeSettings()

8.3 Funcionalitats comunes entre plugins

Totes les extensions disposen de funcions comunes a l'aplicació que poden ser utilitzades. Aquestes funcions són de tipus *static* i *public* per facilitar el seu accés:

- ChatPanel :: public static InetAddress **explorarInterficies()**: Aquesta funció permet explorar tots els dispositius de xarxa (també el loopback) i ens retorna les IP internes que té. En entorns Windows i Mac executa la funció de Java InetAdress.getLocalHost(), però en Linux funciona diferent ja que es va detectar que Java no agafava bé l'adreça local.

- ChatPanel :: public static String **obtenirIPexterna(String direccio)**: En el paràmetre d'entrada cal passar-li una adreça web URL; llavors l'aplicació consulta la pàgina i obté la direcció que li proporciona.

  En la pàgina web **http://triem.udg.edu/plugins/ip.php** s'ha creat un petit script en PHP que retorna la nostra adreça externa a l'extensió.

- ChatPanel :: public static void **setActiveTab(String nompestanya)**: Si li passem el nom de la nostra pestanya, el programa principal la seleccionarà i la mostrarà automàticament per pantalla.

Un altre aspecte que les extensions han de respectar és a on **guardar les dades**. L'aplicació principal ofereix una variable estàtica anomenada ***JBotherLoader.mediaCache***, que conté la ruta a la carpeta on es troben totes les dades de l'usuari (pensant en el cas de les dades en memòries extraïbles).

Exemple pràctic: *String mediaDir = JBotherLoader.mediaCache + File.separatorChar + "mediaStore";*

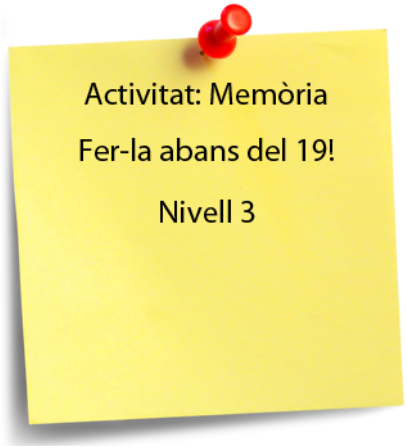
8.4 Implementar activitats asíncrones

Per crear activitats asíncrones s'ha implementat una nova classe anomenada **Activity**, en la qual les extensions la poden instanciar i posar-hi dades. Quan parlem d'una extensió, sol estar **formada per dues parts**: una per l'especialista i l'altra pel pacient. Les dues parts es comuniquen entre elles per poder-se intercanviar informació.

L'especialista és l'encarregat de crear una activitat i d'enviar-la al pacient. Per poder enviar un objecte a través d'un socket en Java, la classe ha de ser *serialitzable*, és a dir, que la codificació de l'objecte es pugui guardar en forma de cadenes de text i enviar-se a través de XML.

Dades que pot emmagatzemar la classe *Activity*:

- Data d'enviament (String)
- Data límit activitat (String)
- Nom de l'activitat (String)
- Activitat completada (Boolean)
- Hashtable de paràmetres (String, String)



Amb la taula de paràmetres (Hashtable), cada activitat pot definir els seus **propis paràmetres**, i en conseqüència la classe *Activity* serveix per qualsevol tipus d'extensió.

Un cop el pacient rep una *Activity*, es guarda en un Vector <*Activity*> i aquest vector s'emmagatzema en un fitxer anomenat *activitats.dat,* que es troba en el perfil de l'usuari. Quan el pacient obre l'aplicació AXARM, aquesta mira si existeix aquest fitxer i, en cas afirmatiu, carrega les activitats.

Un cop el pacient completa una activitat, el programa li envia els resultats a l'especialista de la mateixa manera. L'objecte *Activity* s'emplena amb els resultats de l'activitat en la Hashtable i se li envia a l'especialista com si fos un missatge.

8.4.1 Part de l'especialista

A continuació es mostra un codi per poder crear i enviar una activitat asíncrona.

```
import com.valhalla.jbother.Activity;
import com.valhalla.jbother.ChatPanel;

//Crear una activitat
Activity act = new Activity("Nom_Activitat");
//Afegir atributs
act.afegirParametre("parametre1", valor);
//Posar la data límit del Calendari
act.setDataLimit(cal.getDataAct());
//I enviar el missatge a través de XMPP
```

En canvi, per rebre les activitats és un pèl diferent ja que primer cal fer un recorregut per tot el **vector d'activitats** cercant si hi ha activitats que concorden amb la nostra extensió. El mètode públic ***MessagePacketListener.getExercicis()*** ens retornarà aquest vector.

```
import com.valhalla.jbother.Activity;
import com.valhalla.jbother.jabber.smack.MessagePacketListener;

//Obtenir el llistat d'exercicis rebuts
Vector<Activity> v = MessagePacketListener.getExercicis();
for (int i=0;i<v.size();i++){
  Activity act=v.get(i);
  if (act.getNom().compareTo("Nom_Activitat")==0){ //guardar dades
  }
}
```

8.4.2 Part del pacient

L'aplicació AXARM li avisa al pacient per pantalla de la disponibilitat de nous exercicis i alhora li recorda quines activitats té per completar. Se li presenten en forma de taula, on al fer doble clic en una activitat se li carrega sempre que no estigui ja completada o fora de termini.

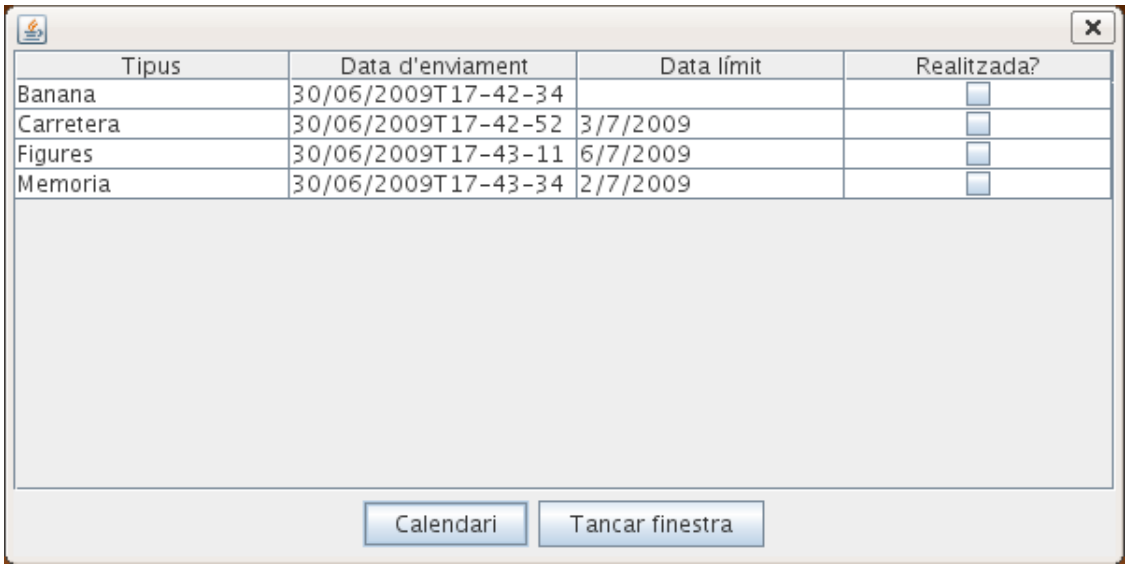


Figura 8.1: Taula on apareix tot l'historial d'activitats del pacient.

El codi per enviar els resultats a l'especialista és similar al descrit en l'apartat anterior, però afegint els resultats. Quan implementem l'extensió del pacient, la funció *MessagePacketListener.getActiveExercice();* retorna l'objecte *Activity* carregat al fer el doble clic.

Un detall molt **important** és que cal marcar la nostra activitat com **completada**, amb el mètode *setDone(true);* i després guardar-la amb el mètode estàtic *MessagePacketListener.saveExercices();*

Un exemple de codi quan finalitzem un exercici seria el següent:

```
import com.valhalla.jbother.Activity;
import com.valhalla.jbother.ChatPanel;
import com.valhalla.jbother.jabber.smack.MessagePacketListener;

//Crear una nova activitat
Activity act = new Activity("Nom_Activitat");
//Afegir resultats
act.afegirParametre("parametre1", valor);
...
//Enviar el missatge a través de XMPP

//Marcar com completada l'activitat REBUDA
activitat.setDone(true);

//I guardar-la en el fitxer d'activitats
MessagePacketListener.saveExercices();
```

Capítol 9

Recursos externs

Al parlar del projecte TRiEM, no només ens referim a l'aplicació AXARM, sinó a una sèrie de recursos que van més enllà de la pròpia aplicació. En aquest capítol farem referència a alguns aspectes que no es poden veure en el programa, però que sens dubte són necessaris per millorar l'experiència dels usuaris.

Pàgines web: Actualment disposem de dues planes web relacionades amb el projecte:

- **http://triem.udg.edu/axarm/** → Aquesta pàgina és mantinguda per la pròpia Universitat de Girona i consta dels següents apartats:
  - Un apartat **d'informació general** del projecte.
  - **Informació tècnica** dels perifèrics provats en el laboratori (càmeres web, joysticks i catifes de ball).
  - Més informació necessària per poder compatibilitzar **Linux** amb la nostra aplicació (instal·lació del joystick, configuració del JMF...)
  - **Vídeos en alta resolució** per visualitzar-los en el propi navegador sobre el funcionament del programa.

  - Un **programa per obtenir les característiques d'un PC** en Windows XP i Vista.
  - Una sèrie de *microsites* molt útils als especialistes a l'hora **d'afegir noves seqüències** en les activitats.

- **http://www.triem.org** → Representa la pàgina institucional del projecte i la més formal. Es pot consultar informació sobre els aspectes més generals del projecte acompanyat d'algunes imatges de l'aplicació.

Figura 9.1: Captura de la pàgina web institucional del projecte.

Volem remarcar sobre els dos últims aspectes esmentats anteriorment de la pàgina TRiEM de la Universitat. El programa per obtenir les característiques d'un PC l'ha desenvolupat una integrant de l'equip del TRiEM (Shaila Jiménez) i ha elaborat un extens manual detallat sobre el seu ús i desenvolupament. Aquest **manual l'hem inclòs en l'apèndix D** i recomanem fer-li una lectura.

Respecte als *microsites*, aquests consisteixen en senzilles planes web per **dissenyar activitats** als pacients. En el moment d'escriure aquestes línies disposem de tres activitats:

- **Catifa de ball**: La web comprova si la seqüència escrita és correcta i la mostra gràficament per saber què haurà de fer el pacient.
- **Figures**: A partir d'una seqüència numèrica, a la web es dibuixen quines figures veurà el pacient.
- **Laberint**: L'especialista dibuixa amb el ratolí la forma d'un laberint i la web li dóna la seqüència escrita per afegir-la a l'aplicació.

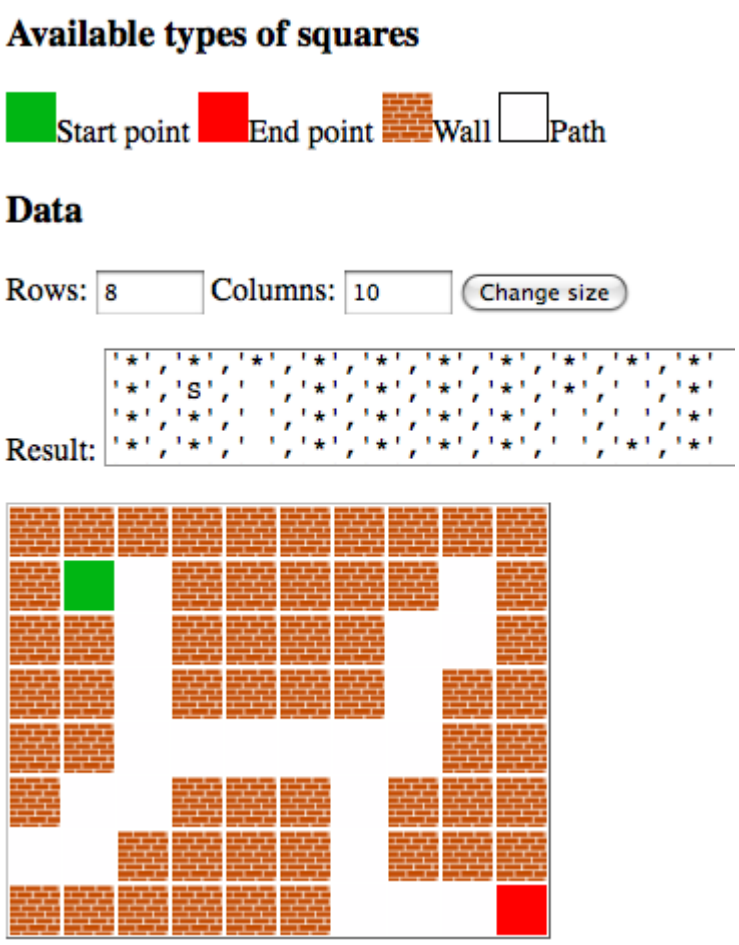


Figura 9.2: Disseny d'un laberint des de la web.

Suport tècnic: En moltes ocasions sorgeixen problemes amb els usuaris (tant especialistes com pacients). Per no haver d'anar a casa de l'usuari cada cop que sorgeix un problema, es fa servir una eina d'assistència remota anomenada **TeamViewer** [33]. El 90% dels problemes es poden solucionar via remota. A més a més, amb aquest sistema també es pot actualitzar l'aplicació manualment.

Instal·lador: Per distribuir fàcilment l'aplicació és necessari fer servir un instal·lador simple i eficaç. Una de les millors opcions a l'hora de crear instal·ladors és el paquet **IzPack** [34]. IzPack és de codi lliure i genera un fitxer JAR autoexecutable. Disposa de moltes opcions de personalització com el suport multiidioma o la compatibilitat amb Windows (XP i Vista), Linux i Mac OS X.

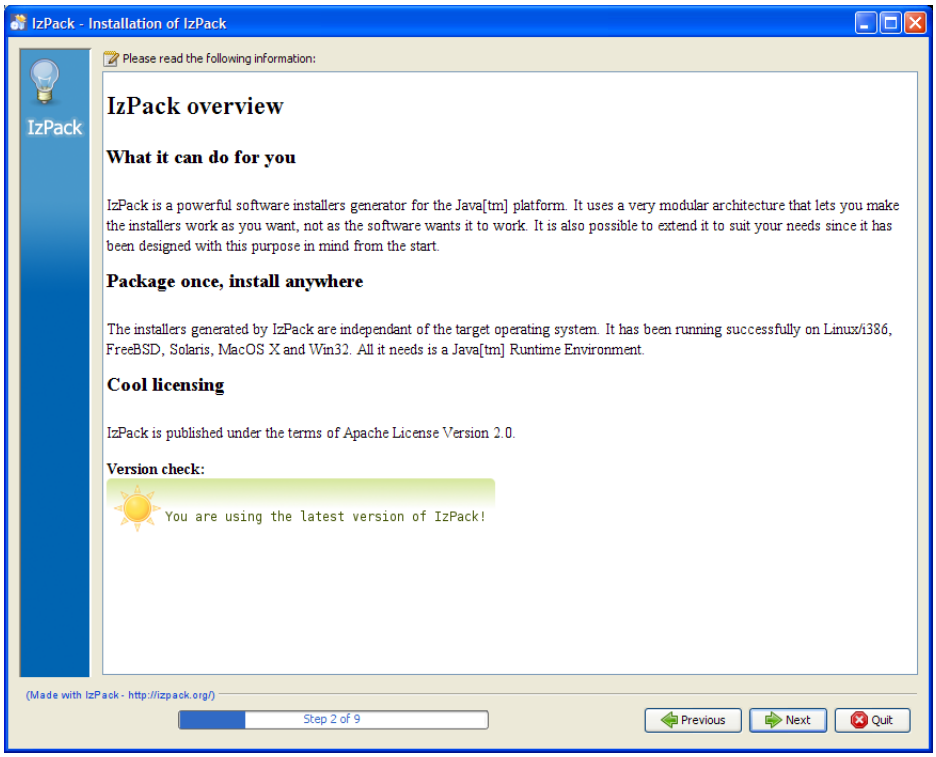


Figura 9.3: Pantalla de l'instal·lador IzPack.

Capítol 10

Proves i resultats

Durant el transcurs d'aquest projecte, la FEM ha mostrat una gran implicació amb l'aplicació AXARM i prova d'això és la voluntat d'aplicar-ho directament als seus pacients. S'ha realitzat una **prova pilot amb sis pacients** de les següents localitats: Vidreres, Girona, Tordera, Banyoles, Palau Saverdera i Riudellots de la Selva. La realitat ens reafirma amb la utilitat de l'aplicació ja que la dispersió geogràfica de la població en la província de Girona hi és present en aquesta mostra.

La primera dificultat més gran va ser la de la instal·lació i configuració dels equips dels pacients. Cada pacient disposava del seu propi ordinador a casa, i en ocasions hi va haver problemes amb ells. Algun ordinador no disposava dels requeriments aconsellables per fer funcionar l'aplicació. D'altres vam haver de fer una neteja i optimització del seu Windows per millorar-ne el rendiment. Fins i tot, en alguna ocasió ens hem trobat amb perifèrics dels ordinadors dels pacients que eren incompatibles amb la nostra aplicació i que ha calgut substituir-los.

Un cop que es va aconseguir fer funcionar l'aplicació en l'ordinador del pacient, llavors calia explicar-li com fer-la anar. A l'igual que amb els ordinadors, hi ha pa-

cients que tenen més rapidesa que d'altres per assimilar-ne el seu funcionament. Per exemple, en alguns casos era necessari explicar el funcionament de les activitats i com havien de fer servir el joystick. Un cop superada la barrera inicial, els especialistes i pacients han pogut fer funcionar l'aplicació perfectament.

Hi ha previst fer un **estudi clínic** al setembre d'aquest any per la FEM amb 30 pacients de la província de Girona (15 pacients amb EM faran servir AXARM i 15 pacients diferents faran teràpia rehabilitadora a l'hospital). Durant un mes els de l'AXARM realitzaran activitats diàriament i els altres aniran 3 cops per setmana a l'hospital.

A banda de les proves, durant la realització del projecte ens han visitant diverses entitats i professionals de la salut i de la informàtica interessats pel projecte TRiEM. Des de professors convidats d'universitats estrangeres fins a psicòlegs, especialistes rehabilitadors i directors de centres de salut. En general, tots ells han mostrat una resposta molt positiva i han vist una gran viabilitat al projecte.

Capítol 11

Conclusions

En aquest projecte s'han explicat diferents tècniques per obtenir una millor **seguretat** en la plataforma TRiEM centrant-se especialment en la part del client. Tampoc s'ha oblidat el tema de la confidencialitat de les dades dels pacients i s'ha ofert una pràctica solució al problema.

Alhora, també s'han tocat diversos aspectes de la **usabilitat**, tant de la pròpia aplicació com en serveis addicionals. Per exemple, s'ha explicat la problemàtica de les actualitzacions i s'ha ofert una solució acceptable per l'usuari.

Cal destacar la possibilitat de crear **activitats asíncrones**, ja que pel pacient li suposa poder combinar la seva vida laboral i personal amb la realització de les activitats encarregades quan li vagi millor. A més a més, tot el procés per generar els resultats i enviar-los a l'especialista és automàtic.

Degut a l'impossibilitat de fer una prova real per provar **l'escabilitat** del sistema, s'ha realitzat una prova local amb un programa de testejar servidors, amb resultats molt satisfactoris.

Volem recordar dues coses del projecte TRiEM: l'aplicació **està funcionant** (no és cap prototipus) i totes les activitats es poden fer de forma **síncrona i asíncrona.**

Finalment, a títol personal, l'esforç que he realitzat ha estat molt positiu i satisfactori, ja que m'ha permès ampliar els meus coneixements d'informàtica. A nivell de coordinació, he treballat en equip amb els altres companys del projecte TRiEM. No només això, sinó que he mantingut una col·laboració constant i necessària tant amb la pròpia Universitat com amb la Fundació Esclerosi Múltiple.

A més a més, treballar en aquest projecte m'ha servit per explorar la meva faceta com a investigador, ja que aquest és un projecte molt lligat a la investigació. No ens hem d'oblidar que el projecte TRiEM té un gran impacte per la seva **aplicabilitat pràctica** dins la societat, de la qual cosa pocs projectes poden presumir-ne. Per tot això, el projecte m'ha estat un complement molt important dels estudis efectuats durant el Màster en Informàtica Industrial i Automàtica.

Capítol 12

Treball futur

Durant el dia a dia en el laboratori del grup BCDS sorgeixen noves idees a explorar. Lamentablement, a vegades per falta de recursos o temps ens és impossible aplicar-les en el mateix moment. Totes aquestes idees es guarden per implementar-les més endavant i algunes d'elles acaben convertint-se en nous projectes finals de carrera.

En un futur immediat ens agradaria aplicar els següents punts:

- Durant dos mesos (Març i Abril de 2009) vam rebre **dos estudiants d'Erasmus** que van realitzar el seu projecte final de carrera participant en el projecte TRiEM. La seva part va consistir en investigar noves possibilitats d'aplicar perifèrics per fer activitats de telerehabilitació d'extremitats superiors i inferiors. El primer projecte [35] es va centrar en aplicar acceleròmetres per fer activitats (detectar moviments de braços). El segon projecte [36] va experimentar amb un dispositiu d'ultrasons, el qual permet mesurar distàncies (un exemple seria veure si el pacient està aixecant el peu i com l'està aixecant).

  Una tasca pendent és completar la seva feina integrant-ho en forma d'extensió més madura, i veure la seva viabilitat.

- A pesar de les millores d'usabilitat, encara queda camí per millorar la interactivitat amb l'aplicació. Per exemple, una possibilitat seria facilitar la generació de noves seqüencies d'activitats pels especialistes, que actualment ho han de fer a través d'una web.

- Hem parlat de fer certificats digitals, però aquests els hem de crear nosaltres mateixos. Ara que cada cop es va incorporant el nou DNI-E a la societat, seria ideal poder-lo fer servir, ja que l'entitat certificadora en aquest cas seria el propi Estat i no la Universitat o la FEM. Només caldria subministrar als pacients un lector de targes de DNI-E (entre 10 i 30€). Hi ha un inconvenient i és que **no es podrien revocar** els certificats.

- En el món dels videojocs, la videoconsola Wii ha suposat una revolució a l'hora de jugar als videojocs. En comptes de fer servir un comandament tradicional, disposa d'un controlador especial anomenat **Wiinote**. El Wiinote és un comandament inalàmbric (per Bluetooth) que incorpora uns acceleròmetres i permet obtenir la posició 3D, velocitat i acceleració d'aquest. Des d'un ordinador podem llegir aquestes dades i aplicar-les a la realització d'exercicis, alhora que té un baix cost (al voltant de 40€).

- Relacionat amb la Wii, existeixen dos perifèrics més que poden oferir molt de joc a les activitats:

  - **Wii Balance Board:** Té la forma d'una taula d'exercicis aeròbics que es col·loca al terra. Disposa de quatre sensors de pressió situats a cada punta de la taula i que permeten mesurar tant la força amb què es pressiona com la posició on es fa la força. Amb només tres sensors, la taula ja podria determinar els càlculs, però segurament s'ha afegit un quart per poder fer comprovacions d'errors. La posició on es fa la força ens permet determinar, per exemple, el centre de gravetat de la persona.

– **Wii Vitality Sensor:** Aquest perifèric s'ha presentat aquest any en l'E3, la fira de videojocs més important de tot el món (2-4 juny). En el moment d'escriure aquestes línies, encara no ha sortit al mercat i es desconeix el seu cost i funcionalitats. Pel que sembla, és un aparell que es col·locarà en el dit i podrà llegir el pols i la tensió de l'usuari. És possible que en un futur pròxim es pugui utilitzar també en un ordinador.

Figura 12.1: A dalt, la Wii Balance Board. A sota, el Wii Vitality Sensor.

Apèndix A

Creació de Certificats amb OpenSSL

Listing A.1: Crear entitat certificadora pròpia (CA).

```
//Crear una carpeta on guardarem tots els certificats i claus
mkdir ./CA
cd ./CA

//Generar CA
openssl req -x509 -newkey rsa:4096 -keyout cakey.pem -days
3650 -out cacert.pem

/*Omplir amb les dades (Pais, Ciutat, Organitzacio...).
El camp CN ha de ser la direccio del servidor!*/

//Eliminar la contrasenya del certificat
openssl rsa -in cakey.pem -out cakey.pem
```

Listing A.2: Crear certificat del servidor a partir de la CA.

```
//Crear clau privada
openssl genrsa -des3 -out serv-priv.pem 4096
openssl rsa -in serv-priv.pem -out serv-priv.pem

//Fer peticio de certificat
openssl req -new -key serv-priv.pem -out petic-cert-serv.pem
//Omplir amb les dades pel servidor

vi config.txt    //Afegir: extendedKeyUsage = serverAuth

//Fer el certificat (acceptar la peticio)
openssl x509 -CA cacert.pem -CAkey cakey.pem -req -in
petic-cert-serv.pem -days 3650 -extfile config.txt -sha1
-CAcreateserial -out servidor-cert.pem
```

Disposem dels següents fitxers:

- CA: cacert.pem, cacert.srl i cakey.pem
- Servidor: servidor-cert.pem i serv-priv.pem

Per fer-ho servir amb el servidor de XMPP, cal ajuntar els dos fitxers del servidor (certificat+clau privada) en un de sol, i indicar-li en la configuració del server.
Ara, amb la CA es poden crear tants certificats de client com facin falta.

Listing A.3: Crear certificats clients.

```
//Clau privada cleint
openssl genrsa -des3 -out client-priv.pem 4096
openssl rsa -in client-priv.pem -out client-priv.pem

//Fer peticio de certificat
openssl req -new -key client-priv.pem
-out petic-cert-client.pem
//Omplir amb les dades del client, el CN pot ser el JID

vi config1.txt    //Afegir: extendedKeyUsage = clientAuth

//Fer el certificat (acceptar la peticio)
openssl x509 -CA cacert.pem -CAkey cakey.pem -req -in
petic-cert-client.pem -set_serial 3 -days 3650
-extfile config1.txt -sha1 -out client-cert.pem
```

És important el camp **set_serial** perquè indica quin número de certificat es crea. Cada cop que en fem un de nou, cal anar augmentant aquesta xifra. Pel primer client és el 3, ja que l'1 és la CA i el 2 el servidor.

Disposem ara dels següents fitxers:

- CA: cacert.pem, cacert.srl i cakey.pem
- Servidor: servidor-cert.pem i serv-priv.pem
- Client: client-cert.pem i client-priv.pem

Es poden eliminar tots els altres fitxers generats (els config.txt, i les peticions).

Listing A.4: Opcional: exportar a PKCS12 per navegadors webs.

```
openssl pkcs12 -export -in client-cert.pem -inkey
client-priv.pem -certfile cacert.pem -out cert-pck12.p12
//Demana generar una contrasenya
```

Listing A.5: Crear Llista de revocació de certificats.

```
//Primer cal editar la configuracio de OpenSSL
sudo vi /etc/ssl/openssl.conf
//En l'apartat [CA_default] editar...
dir = "On es troba la carpeta CA"
#crl_dir = ...
#unique_subject = no
certificate = $dir/cacert.pem
#crlnumber
#crl
private_key = $dir/cakey.pem
RANDFILE = $dir/.rand

//Generar la llista de certificats a revocar
openssl ca -gencrl -out llistarev.crl

//Convertirla a format .DER pel Openfire
openssl ca -gencrl -out llistarev.pem
openssl crl -in llistarev.pem -out llistarev.der
-sha1 -outform DER
```

Listing A.6: Treballar amb revocació.

```
//Per revocar un certificat
openssl ca -revoke client-cert.pem


//Per mostrar tots els que hi ha:
openssl crl -in llistarev.crl -text -noout
```

Cada cop que revoquem un certificat cal **regenerar** la llista dels certificats revocats (tornar a cridar -gencrl). En el servidor Openfire se li indica on es troba el fitxer crl.

Listing A.7: Opcional: Llistar certificats que fa servir un servidor XMPP.

```
//Cal utilitzar el port 5223 = protocol antic SSL
openssl s_client -connect servidor:5223 -showcerts
```

Apèndix B

Fitxer configuració XML Tsung

```
1  <?xml version="1.0"?>
2  <!DOCTYPE tsung SYSTEM "/usr/share/tsung/tsung-1.0.dtd">
3  <tsung loglevel="notice" version="1.0">
4  <clients>
5      <client host="localhost" use_controller_vm="true"></client>
6  </clients>
7
8  <servers>
9      <server host="songohan.udg.edu" port="5222" type="tcp"></server>
10 </servers>
11
12 <load>
13     <arrivalphase phase="1" duration="1" unit="minute">
14         <users interarrival="0.01" unit="second"></users>
15     </arrivalphase>
16
17     <arrivalphase phase="2" duration="5" unit="minute">
18         <users interarrival="0.1" unit="second"></users>
19     </arrivalphase>
20
21     <arrivalphase phase="3" duration="3" unit="minute">
```

```
22          <users interarrival="1" unit="second"></users>
23      </arrivalphase>
24  </load>
25  <!-- JABBER parameters -->
26  <options>
27      <option type="ts_jabber" name="global_number" value="300"></option>
28      <option type="ts_jabber" name="userid_max" value="1000"></option>
29      <option type="ts_jabber" name="domain" value="songohan.udg.edu"></
            option>
30      <option type="ts_jabber" name="username" value="tsung"></option>
31      <option type="ts_jabber" name="passwd" value="p4ssw0rd"></option>
32  </options>
33
34  <sessions>
35      <!-- for each user we do the same -->
36      <session probability="100" name="jabber-example" type="ts_jabber">
37          <!-- connect and logon -->
38          <request> <jabber type="connect" ack="no_ack"></jabber> </
                request>
39          <thinktime value="2"></thinktime>
40
41          <transaction name="authenticate">
42              <!-- ack=global:wait for all other users to authenticate -->
43              <request> <jabber type="auth_get" ack="global"></jabber> </
                    request>
44              <request> <jabber type="auth_set_plain" ack="local"></jabber
                    > </request>
45          </transaction>
46
47          <request> <jabber type="presence:initial" ack="no_ack"/> </
                request>
48          <thinktime value="2"></thinktime>
```

```
49
50          <transaction name="roster">
51              <request> <jabber type="iq:roster:get" ack="local"></jabber>
                    </request>
52          </transaction>
53
54          <thinktime value="5"></thinktime>
55
56          <transaction name="online">
57              <request> <jabber type="chat" ack="no_ack" size="16"
                    destination="online"></jabber> </request>
58          </transaction>
59          <!-- add another user to the roster; delete him later -->
60          <transaction name="rosteradd">
61              <request> <jabber type="iq:roster:add" ack="no_ack"
                    destination="online"></jabber> </request>
62              <request> <jabber type="presence:subscribe" ack="no_ack"/> <
                    /request>
63          </transaction>
64
65          <thinktime value="1"></thinktime>
66          <!-- write some message to other online and offline (this will
                cause some errors) users -->
67          <transaction name="online">
68              <request> <jabber type="chat" ack="no_ack" size="56"
                    destination="online"></jabber> </request>
69          </transaction>
70
71          <thinktime value="4"></thinktime>
72
73          <transaction name="online">
74              <request> <jabber type="chat" ack="no_ack" size="16"
```

```
                destination="online"></jabber> </request>
75          </transaction>
76
77          <transaction name="rosterrename">
78              <request> <jabber type="iq:roster:rename" ack="no_ack"></
                    jabber> </request>
79          </transaction>
80
81          <thinktime value="3"></thinktime>
82
83          <transaction name="offline">
84              <request> <jabber type="chat" ack="no_ack" size="56"
                    destination="offline"></jabber> </request>
85          </transaction>
86
87          <thinktime value="3"></thinktime>
88
89          <transaction name="rosterdelete">
90              <request> <jabber type="iq:roster:remove" ack="no_ack"></
                    jabber> </request>
91          </transaction>
92          <!-- and disconnect -->
93          <transaction name="close">
94              <request> <jabber type="close" ack="no_ack"></jabber> </
                    request>
95          </transaction>
96      </session>
97  </sessions>
98  </tsung>
```

Apèndix C

Llistat dels components que fa servir AXARM

L'aplicació AXARM fa servir, com moltes altres aplicacions actuals, llibreries i components per realitzar algunes tasques. A continuació s'enumera un llistat de tots els components que utilitzen tant l'aplicació com les extensions desenvolupades, i amb la seva llicència.

A) Nucli de l'aplicació

- Java 6 → (Binary Code Licence Agreement): Permet distribuir el software fet en Java i la màquina virtual sense cost, sempre que no es modifiqui el Java i l'usuari final accepti la llicència.
- Jbother 0.8.9 → GNU GPL v1
- Components utilitats en JBother:
  - dotuseful Library → Apache License 1.1
  - JCommon → LGPL 2.1 (GNU Lesser General Public License)
  - Smack 3.1.0 → Apache License 2.0

B) Extensions o *Plugins*

Llibreries comunes a vàries extensions

- jinput → BSD Licence (Berkeley Software Distribution)
- j2ssh 0.2.9 (sshtools) → GNU GPL v2
- json (Javascript Object Notation) → Domini públic
- jzlib → BSD Licence

Videoconferència

- JMF → Llicència pròpia (el codi font té llicència SCSL - Sun Community Source Licensing)
- FOBS4JMF → LGPL
- media4j 0.1.4 → BSD Licence

Bloc de notes i Llibreria Multimèdia

- ekit 1.3 → LGPL 2.1
- jfreechart → LGPL 3.0

Altres extensions en desenvolupament

- CarRace codi font: → GNU GPL v3 (amb consentiment de l'autor)
- Backtracking codi font: → GNU GPL v3 (amb consentiment de l'autor)
- jd2xx → BSD Licence

C) Imatges

- Clker.com → domini públic
- iconspedia.com → depèn de l'autor, solen ser gratuïtes
- iconarchive.com → depèn de l'autor, solen ser gratuïtes

D) Icones

- famfamfam → Creative Commons Attribution 2.5 o 3.0
- Tango Desktop Project → GNU GPL v3
- Project Nuvola → LGPL

E) Sons

- freesound.org → Creative Commons Sampling Plus 1.0

F) Externs a l'aplicació

http://triem.udg.edu/axarm

- jquery → MIT (Massachusetts Institute of Technology) o GNU GPL
- jquery-draw → GNU GPL v2

Programa per detectar hardware

- AutoHotKey → GNU GPL v2
- Info-ZIP 2.3 → BSD Licence

Instal·lador

- Izpack 4.3.0 → Apache License 2.0

Utilitzar càmeres V4L2 en entorns Linux

- Flashcam Project → GNU GPL v2

Generar claus RSA i xifrar dades

- openssl → Apache License
- TrueCrypt → TrueCrypt License 2.6 (Open source)

Generar stress tests

- Tsung 1.3.0 → GNU GPL v2

G) Servidors (XMPP, Web, SFTP)

- OpenFire 3.6.4 → GNU GPL v2
- ejabberd 2.0.5 → GNU GPL v2
- Apache Web Server → Apache License 2.0
- mysecureshell 1.15 → GNU GPL v2

Apèndix D

Test PC: Programa propi per llegir les característiques d'un PC

D.1 Introducció

El ***Test PC*** del suport tècnic és una eina per recol·lectar informació sobre el maquinari, sistema operatiu i aplicacions instal·lades dels ordinadors dels pacients. La finalitat del test és saber si l'ordinador compleix els requisits mínims per la instal·lació i el correcte funcionament de l'AXARM. El test únicament obté la informació necessària perquè els tècnics avaluïn si un ordinador és apte o no per suportar l'AXARM, en cap cas es recopila ni es distribueix informació personal del pacient.

La informació recollida és enviada a través d'Internet a una màquina remota mitjançant el protocol de transferència de fitxers (FTP) al personal de suport tècnic.

El test permet que el suport tècnic, d'una manera ràpida, tingui el coneixement del maquinari que disposa el pacient abans d'anar al seu domicili i pugui anticipar-se a possibles incompatibilitats i problemes.

No és una aplicació que queda instal·lada permanentment a l'ordinador del pacient ni tampoc és un servei del sistema: només s'ha d'executar una sola vegada.

D.2 Requisits

- Un ordinador.
- Connexió a Internet (ADSL o equivalent).
- Un sistema operatiu Windows instal·lat a l'ordinador (XP, VISTA, 2000, 2003). No és compatible amb Linux o Mac OS X.

D.3 Informació General

La informació que recull el test es divideix en dues famílies: maquinari i programari.

En quant a maquinari necessitem saber el processador, la memòria i la targeta gràfica. Referent al programari, necessitem conèixer quines aplicacions té instal·lades per detectar possibles problemes, com és el cas de múltiples màquines virtuals de Java instal·lades en el mateix sistema operatiu.

Si el pacient disposa de càmera per la videoconferència, sabrem si té els controladors instal·lats correctament. Podria donar-se el cas que un pacient hagués utilitzat anteriorment altres càmeres i tingués controladors antics sense eliminar, factor que influeix negativament en la detecció de la càmera en el Java Media Framework (JMF).

El llistat de la informació que recull el test és la següent:

- **Sistema Operatiu**
  - Versió del sistema operatiu de la família Windows, Service Pack (si n'hi ha d'instal·lats).
- **Processador**
  - Model, Velocitat, Nombre de *CPU's*.
- **Memòria**
  - Nombre de bancs de memòria, Quantitat de memòria de cada banc.
- **Targeta gràfica**
  - Fabricant, Model, Memòria de Vídeo, integrada o dedicada.
- **Unitats d'emmagatzematge**
  - Espai disponible al disc principal.
- **Dispositius PnP (Plug and Play)**
  - Càmera Web, Joystick, Dispositius de xarxa inalàmbrics.
- **Dispositius de xarxa**
  - Targetes de xarxes.
- **Aplicacions instal·lades**
  - Versions prèvies del Java, Controladors dels dispositius PnP.

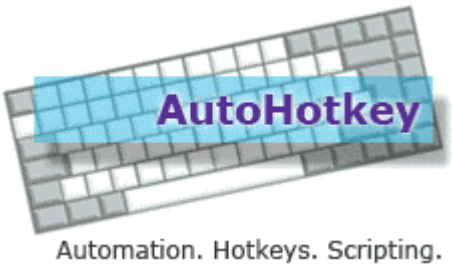
D.4 Desenvolupament del script

El desenvolupament del test s'ha realitzat a partir d'un script en llenguatge **AutoHotkey** [37]. Per tal d'assegurar la portabilitat a altres sistemes s'ha compilat el script en un fitxer executable.

D.4.1 Què és AutoHotkey?

És una utilitat lliure i gratuïta per a Windows que permet fer el següent:

- Automatitzar tasques enviant accions del teclat i/o clics de ratolí. També es poden escriure *macros*.
- Crear hotkeys pel teclat, ratolí i joystick.
- Crear interfícies, menús i barres.
- Redefinir tecles i botons del teclat, ratolí i joystick.
- Convertir qualsevol script en un fitxer executable en ordinadors que no tinguin l'AutoHotkey instal·lat.
- Afegir altres executables al compilat final.



Gràcies a les funcionalitats que ens proporciona treballar amb l'AutoHotkey, la intervenció humana del pacient és quasi nul·la.

El disseny del programa consta de tres fases: **generació de dades, compressió i enviament.**

En la primera fase de recollida, s'extreuen del sistema totes aquelles dades que són d'interès i es guarden en un o més fitxers de text.

En la següent fase de compressió, els fitxers de text que contenen la informació es comprimeixen en un únic fitxer perquè sigui més fàcil i ràpid el seu enviament a través d'Internet. Per a cada pacient, el fitxer es guarda amb un identificador (nom del pacient i la data/hora de l'enviament).

En l'última fase d'enviament, el fitxer viatja a través d'Internet fins arribar a un servidor FTP que està al laboratori del BCDS. Posteriorment, els tècnics accedeixen al servidor i revisen els resultats en busca de conflictes per recomanar-li solucions.

D.4.2 Fase I: Generació dades. msinfo32

La utilitat que s'encarrega de generar una llista detallada amb la configuració del computador és el **msinfo32**, un executable que està integrat en tots els sistemes operatius de la família Windows. Es troba en la ruta *%commonprogramfiles%\Microsoft Shared\MSInfo.*

Ens permet crear un o varis fitxers *.nfo o .txt* que contindran informació corresponent a categories especifiques.

Per evitar que els usuaris puguin tenir aquest fitxer malmès o amb una versió diferent, afegim l'executable **msinfo32** a dins del programa final, ja compilat, perquè sigui portable a tots els ordinadors.

D.4.3 Fase II: Compressió. Info-zip

En la fase prèvia de generació de dades, s'ha guardat en un directori específic un o varis fitxers *.nfo*. Tot i que els fitxers resultants de **msinfo32** són de mida reduïda, si es realitza l'enviament de varis fitxers resulta més beneficiós si és un únic fitxer.

Per aquest motiu, els informes es comprimeixen amb extensió *.zip* que serà de mida molt més reduïda i permetrà enviar les dades amb més agilitat.

Per a la compressió s'utilitza **Info-zip** (una eina de codi lliure) que s'afegeix a la versió compilada del programa.

Per reduir al màxim el fitxer resultant s'utilitza la compressió màxima (nivell 9). Tot i així, la mida del fitxer depèn de la quantitat d'informació que ha recollit el msinfo32. En general, els fitxers comprimits són de mida reduïda i de l'ordre de KB.

D.4.4 Fase III: Enviament. Utilitat FTP Windows

Després de comprimir els resultats, aquests s'han de transferir fins a una màquina remota de forma ràpida. És per això que s'utilitza el protocol de xarxa FTP basat en arquitectura client-servidor.

Des de l'equip del pacient el test realitza una connexió al servidor, ubicat a les instal·lacions del BCDS, per enviar el fitxer comprimit i desar-lo.

Existeixen diverses solucions de software que permeten utilitzar el protocol FTP per la transferència de fitxers, però en el nostre cas aprofitarem el client FTP que està integrat al Windows.

És important saber com s'han de transportar les dades al llarg de la xarxa ja que sinó es pot destruir la informació del fitxer. Per això quan s'executa l'aplicació FTP és important remarcar que enviarem un fitxer comprimit ZIP (binari) i que cal activar la opció "binary".

En la part del servidor s'ha creat un usuari que només té permisos d'escriptura en el directori (per exemple, no pot llistar directoris). L'adreça del servidor, les dades de login i les ordres que es volen executar en el servidor estan emmagatzemades en un fitxer de text. Aquest fitxer es passa com a paràmetre d'entrada a l'aplicació FTP per realitzar la tasca. Per raons de seguretat, el fitxer de text amb totes les dades del FTP es destrueix abans de tancar l'aplicació.

Cal considerar el perill de que s'empleni el servidor de dades, però gairebé seria considerat com una gamberrada degut a les moltes contramesures existents: limitar la cuota del servidor, bannejar una IP...

Per realitzar l'enviament és necessari que l'ordinador es trobi connectat a Internet. El test realitza una comprovació de la xarxa i si no existeix connexió, avisa a l'usuari mitjançant un missatge d'error per pantalla i tanca el test. Pel contrari, si el test detecta que la connexió establerta a Internet és correcta, prossegueix amb l'enviament del fitxer.

D.5 Interacció amb l'usuari

El test avisa de la seva presència i el seu progrés durant tota l'execució mitjançant uns missatges en forma de globus informatius que van apareixent a la barra de tasques de Windows. Els globus informen al pacient sobre l'estat del test.

Si es produeix algun error durant l'execució, s'avisa a l'usuari amb un globus de color vermell i es tanca el test.

Si en qualsevol moment durant l'execució, el pacient decideix que vol parar el test, només cal que premi la *tecla ESC* i el test demanarà la seva confirmació. En cas afirmatiu, el test es tancarà sense haver enviat les dades. Si durant 15 segons el test no rep la confirmació, té l'ordre de reiniciar la seva execució de nou.

Abans del tancament de l'aplicació s'inicia el procés de neteja. S'esborren tots els fitxers que s'han anat generant (fitxers .nfo i .zip, fitxer d'ordres FTP, executables msinfo32.exe i zip.exe) mentre que a l'usuari li apareix una barra progressiva.

D.5.1 Detalls tècnics

- Es crea un directori de treball a l'Escriptori. En aquesta ubicació no hi ha problemes de permisos d'escriptura.
- El directori de treball té un nom únic per evitar que a la fase de neteja s'esborrin dades de l'usuari.
- Funciona tant en comptes d'usuari limitats, com en comptes d'administrador.
- Per comprovar si l'usuari ha apretat o no la tecla ESC, es programa un temporitzador que consulta l'estat de la tecla cada 10 mil·lisegons.

- Per guardar les ordres del servidor es crea un nou fitxer de text on cada línia representa una ordre al servidor FTP.
- El test queda minimitzat a la barra de tasques, representat per una icona.
- Els detalls sobre l'estat de la xarxa es poden saber mitjançant una consulta a "wininet.dll".

Bibliografia

[1] E. Vasilyeva, M. Pechenizkiy, and S. Puuronen. Towards the framework of adaptive user interfaces for ehealth. In *Proc. 18th IEEE Symposium on Computer-Based Medical Systems*, pages 139–144, 23–24 June 2005.

[2] Universitat de Girona. BCDS - Broadband Communications and Distributed Systems. http://bcds.udg.edu.

[3] Fundació Esclerosi Múltiple. http://www.fem.es.

[4] A. Olsen. JBother - A Groovy Jabber Client. http://www.jbother.org.

[5] XMPP Standards Foundation. XMPP - eXtensible Messaging and Presence Protocol. http://www.xmpp.org.

[6] IETF. IETF - Internet Engineering Task Force. http://www.ietf.org.

[7] Sun Microsystems, Inc. JMF - Java Media Framework API. http://java.sun.com/javase/technologies/desktop/media/jmf.

[8] K. Larson, et al. FMJ - Freedom for Media in Java. http://fmj-sf.net.

[9] Generalitat de Catalunya - Departament de Salut - CatSalut. AATRM - Agència d'Avaluació de Tecnologia i Recerca Mèdiques. http://www.gencat.cat/salut/depsan/units/aatrm/html/ca/Du8/index.html.

[10] JM. Tormos Muñoz, EJ. Gómez, A. García-Molina, E. Opisso, and R. Maspons. *Análisis del estado actual de los servicios de telemedicina enfocado a evaluar la viabilidad de un programa de telerrehabilitación en pacientes con una gran discapacidad de origen neurológico*. Number 2006/11. Madrid: Plan Nacional para el Sistema Nacional de Salud del Ministerio de Sanidad y Consumo. Barcelona: Agència d'Avaluació de Tecnologia i Recerca Mèdiques., 2007.

[11] AVANTE - 1er saló per a l'Autonomia Personal i la Qualitat de Vida. http://www.salonavante.com/.

[12] C. Guadall. *Triem: a multiple sclerosis telerehabilitation application*. 2007. [Recurs electrònic] /Carles Guadall Blancafort ; director/tutor: José Luis Marzo Lázaro i Antonio Bueno Delgado; disc òptic (CD-ROM; PFC-EPS, Enginyeria Informàtica (2n cicle) – Universitat de Girona, 2007).

[13] X. Vallejo. *AXARM : una aplicació eXtensible per assistència remota i monitorització*. Universitat de Girona, Girona, 2008. [Recurs electrònic] /Xavier Vallejo López ; tutor: Antonio Bueno Delgado; 21 gener 2009; Consultable des del DUGI; PFC-EPS, Enginyeria Informàtica (2n cicle).

[14] A. Bueno, C. Guadall, J. Marzo, and D. Harle. TRiEM: An application for MS TeleRehabilitation. In *EDAS 1569053200; eMedisys 2007; First International Conference of E-Medical Services Fez, Morocco, October 24-26, 2007*, 2007.

[15] A. Bueno, J. Marzo, and X. Vallejo. AXARM: An Extensible Remote Assistance and Monitoring Tool for ND Telerehabilitation. In *Electronic Healthcare, First International Conference, eHealth 2008, London, UK, September 8-9, 2008. Revised Selected Papers*, pages 106–113, 2008.

[16] Universitat de Girona. TRiEM: Recerca de la UdG per a l'esclerosi múltiple. *Engega*, 6:27–28, Maig 2008.

[17] Frank Niedermann. Open-source XMPP server comparison chart. http://www.saint-andre.com/jabber/jsc.

[18] Ignite Realtime. Openfire Server. http://www.igniterealtime.org/projects/openfire.

[19] Ignite Realtime. Smack API. http://www.igniterealtime.org/projects/smack.

[20] ProcessOne. ejabberd - High Performance Instant Messaging. http://www.process-one.net/en/ejabberd.

[21] Xiaoka. jabberd - XMPP Server. http://codex.xiaoka.com/wiki/jabberd2:start.

[22] Artur Hefczyc. tigase.org | Open Source and Free (GPLv3) Jabber/XMPP environment. http://www.tigase.org.

[23] Mozilla Foundation. Mozilla Firefox. http://www.mozilla.com/firefox.

[24] Ministerio de Economia y Hacienda. Fábrica Nacional de Moneda y Timbre. http://www.fnmt.es.

[25] Àgencia Catalana de Certificació. CATCert. http://www.catcert.cat.

[26] The OpenSSL Project. OpenSSL: The Open Source toolkit for SSL/TLS. http://www.openssl.org.

[27] Ignite Realtime. Openfire+Smack - Client X.509/PKI Certificate Support. http://www.igniterealtime.org/community/message/192019.

[28] AgentBob. Import private key and certificate into Java Key Store (JKS). http://www.agentbob.info/agentbob/79-AB.html.

[29] Gobierno de España. Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal. *Boletín Oficial del Estado*, 298:43088–43099, Decembre 1999.

[30] TrueCrypt Foundation. TrueCrypt - Free Open-Source Disk Encryption. http://www.truecrypt.org.

[31] TrueCrypt User's Guide. Technical report, TrueCrypt Foundation, 2009.

[32] ProcessOne. Tsung. http://tsung.erlang-projects.org.

[33] TeamViewer. TeamViewer - Free Desktop Sharing and Remote Control. http://www.teamviewer.com.

[34] Julien Ponge. IzPack - Package once. Deploy everywhere. http://izpack.org.

[35] W. Vanderhoydonk. *Research and development of peripherals for upper and lower extremities tele-rehabilitation.* 2009. director/tutor: Antonio Bueno Delgado i José Luis Marzo Lázaro; disc òptic (CD-ROM; PFC-EPS, Enginyeria Tècnica en Informàtica de Sistemes – Universitat de Girona, 2009).

[36] F. van der Have. *Research and development of peripherals for upper and lower extremities tele-rehabilitation.* 2009. director/tutor: Antonio Bueno Delgado i José Luis Marzo Lázaro; disc òptic (CD-ROM; PFC-EPS, Enginyeria Tècnica en Informàtica de Sistemes – Universitat de Girona, 2009).

[37] AutoHotkey. AutoHotkey - Free Mouse and Keyboard Macro Program. http://www.autohotkey.com.

Índex de figures

1.1 L'esclerosi múltiple afecta i danya al sistema nerviós del cos humà. . . 4

1.2 Imatge de l'aplicació en que es veu la interacció entre usuaris. 8

1.3 Logotip del programa AXARM. 9

1.4 Arquitectura híbrida utilitzada en el TRiEM. 10

1.5 Principals companyies que recolzen el XMPP. 11

2.1 Logotip de la fira AVANTE. 16

2.2 Planificació de tasques realitzades durant el projecte. 20

3.1 Panell de control del servidor Openfire. 22

3.2 Panell de control del servidor ejabberd. 23

3.3 Panell de control del servidor Tigase. 24

4.1 L'aplicació AXARM senyalant la part d'Estat del Sistema. 27

4.2 El calendari de l'especialista amb el dia actual (18) i una data triada (21). 29

4.3 Activitat de la memòria. 31

4.4 Activitat de la carretera. 32

4.5 Activitat de la macedònia. 33

4.6 Activitat utilitzant una catifa de ball. 34

4.7 Activitat on el pacient ha de memoritzar una figura geomètrica per reproduir-la a continuació. 35

5.1 Imatge del nou DNI Electrònic extreta de la web www.dnielectronico.es 37

6.1 Mostrant a l'usuari que estem cercant dispositius extraïbles. 52
6.2 Logo del programa TrueCrypt. 53
6.3 Diàleg que surt al posar el pendrive en un Windows. 57
6.4 Diàleg que surt al posar el pendrive en un Ubuntu. 58
6.5 Estructura de la memòria preparada pels tres sistemes operatius. . . . 59

8.1 Taula on apareix tot l'historial d'activitats del pacient. 83

9.1 Captura de la pàgina web institucional del projecte. 86
9.2 Disseny d'un laberint des de la web. 87
9.3 Pantalla de l'instal·lador IzPack. 88

12.1 A dalt, la Wii Balance Board. A sota, el Wii Vitality Sensor. 95