

1. Introducció

Cada vegada més, les empreses s'interessen en la col·laboració d'obres socials per tal de millorar la seva imatge de cara als clients. La responsabilitat social corporativa, o també anomenada responsabilitat social empresarial, pot definir-se com a la contribució activa i voluntària de la millora social, econòmica i ambiental per part de les empreses, generalment, amb l'objectiu de millorar la seva situació competitiva i valorativa i el seu valor afegit. Així doncs, aquesta contribució social no deixa de ser un interès per tal d'obtenir els màxims beneficis.

Des d'un punt de vista personal, l'interès per col·laborar en una obra social implica un respecte com a deute en front la societat a més a més d'una realització i satisfacció personals. D'aquesta manera, una petita aportació de qualsevol tipus, no només de caire econòmic, per part de cadascú permet grans beneficis socials i millores en la qualitat de vida de moltes persones que realment necessiten atenció especial.

Les fundacions sense afany de lucre que contribueixen a iniciatives socials disposen de recursos limitats per la quantitat de necessitats i demandes que se l'hi ofereixen. El fet de realitzar un estudi o projecte pot ser una bona oportunitat per col·laborar en un iniciativa social i així poder facilitar el dia a dia que per algunes persones resulta complicat.

L'oferta d'una oportunitat com aquesta, en la qual es veu una utilitat a una feina desenvolupada i, a més a més, que ajudi a les persones més necessitades comporta un gran al·licient a l'hora d'afrontar-lo, treballar-hi i finalitzar-lo amb bons resultats.

2. La Fundació

La Fundació privada El Maresme Pro Persones Amb Disminució Psíquica és una institució d'iniciativa social sense afany de lucre que promou i impulsa la integració social de les persones amb disminució psíquica des dels principis de normalització i reconeixement de la diferència.

La Fundació el Maresme va ser constituïda el 22 de desembre de 1994, continuant la tasca iniciada l'any 1966 per l'Associació Patronat Pro Persones Amb Disminució Psíquica del Maresme.

Com a centre comarcal de referència, atén i dona resposta a qualsevol qüestió relacionada amb les persones amb disminució psíquica, ja sigui a nivell informatiu i d'orientació, o d'assistència o tractament.

La Fundació el Maresme procura donar resposta a les necessitats i demandes de la comarca, organitzant una xarxa de serveis que ofereixin una atenció en continuïtat a la persona amb disminució psíquica i la seva família.

Per tal de donar servei a les diferents necessitats, la Fundació es troba distribuïda en departaments:

Centre de Desenvolupament Infantil i Atenció Precoç del Maresme (CDIAP)

És un servei de diagnòstic i tractament en règim ambulatori, completament gratuït per les famílies, que atén als infants des dels 0 als 6 anys que presenten algun tipus de trastorn, disminució, disfunció o disharmonia en el seu desenvolupament, o que es troben en situació de risc de patir-la.

Escola d'educació Especial l'Arboç

Atén nens i nenes afectats de paràlisis cerebral o altres lesions amb manifestació motora que interfereixen greument en el procés de desenvolupament i d'aprenentatge.

Servei Ocupacional d'Inserció (SOI)

El SOI és un Centre Ocupacional que acull persones adultes, a partir dels 18 anys, amb un grau de disminució psíquica igual o superior al 33% i amb un nivell de quocient intel·lectual entre mig i lleuger, que han acabat el període de formació escolar, però que encara no estan integrats dins del Centre Especial de Treball ni dins del Servei de Teràpia Ocupacional.

Centre Especial de Treball del Maresme (CEO)

El CEO del Maresme té com a objectiu la integració laboral, personal i social de la persona amb discapacitat psíquica, en règim de treball normalitzat, amb edat laboral (a partir dels 16 anys), ja sigui a l'empresa ordinària o dins del propi Centre Especial de Treball.

Dóna resposta laboral a persones amb discapacitat psíquica, a partir dels 16 anys, amb un grau de disminució psíquica igual o superior al 33% i capacitat productiva.

El Centre Especial de Treball és en l'actualitat una empresa que ofereix treball a unes 200 persones amb discapacitat psíquica i que continua la seva tasca de recerca de llocs de treball per donar una possibilitat de sortida laboral a les persones que actualment estan en formació.

Servei de Teràpia Ocupacional (STO)

Els Serveis de Teràpia Ocupacional (STO) són l'alternativa des de l'àmbit de serveis socials a la integració de les persones amb disminució psíquica que no poden incorporar-se al sistema de treball ordinari per no arribar a un determinat nivell de productivitat.

Atén persones adultes, a partir de 18 anys, amb un grau de disminució del 65% o més, i amb un nivell entre mig i sever de dependència. L'objectiu principal és afavorir al màxim la integració de les persones amb disminució psíquica mitjançant una atenció diürna de tipus rehabilitador integral.

Servei d'Habitatges

El Servei d'Habitatges té per objecte facilitar, de forma temporal o permanent, una vida independent i equivalent a la pròpia llar a persones adultes, amb una disminució psíquica en grau igual o superior al 33%.

Servei d'Acompanyament i Suport a la Vida Independent (SASVI)

El Programa de suport a l'autonomia a la pròpia llar pretén proporcionar a aquelles persones amb disminució psíquica amb un nivell alt d'autonomia i capacitats, una sèrie de suports puntuals i intermitents, que els facilitin la seva integració social, els permetin viure de manera autogestionada i portar una vida independent i desinstitucionalitzada al seu domicili particular, ja sigui de lloguer o en propietat, sols, o acompanyats d'altres persones beneficiàries d'aquest programa.

L'atenció i suport en la vida independent fomenta l'autonomia com a eina del desenvolupament personal i social, i actua complementàriament a d'altres mitjans d'intervenció i xarxes de suport com la comunitat, família o amics.

Residència Llar Sta. Maria

La residència i CAE Llar Sta. Maria Assumpta és un establiment de la Fundació Maresme, adreçat a persones amb disminució psíquica greu en edat adulta, que degut al seu alt grau d'afectació necessiten assistència d'una forma bàsica i continuada i que, per raons familiars i socials, no poden viure a casa seva.

3. Proposta de continuació del Messenger Visual

3.1 Messenger Visual i SPC

Les persones discapacitades tenen moltes i importants dificultats d'adaptació amb les noves tecnologies de comunicació, el que redueix la seva autoestima i autosuficiència, i incrementa el risc de patir una exclusió social i digital. El *Messenger Visual* és un innovador servei de missatgeria instantània basat en icones amb l'objectiu de millorar la comunicació entre persones amb discapacitats que tenen dificultats de lectura o escriptura, d'expressió i de moviment. El Messenger Visual utilitza el sistema SPC (símbols criptogràfics de comunicació), que representa paraules i conceptes a través de criptogrames i dibuixos.

El sistema SPC de comunicació està format per una col·lecció de dibuixos o pictogrames que formen un sistema de comunicació desenvolupat a partir de la dècada dels 80 per Mayer i Johnson. En els pictogrames hi trobem accions, vocabulari, adjectius... que són un suport pels alumnes amb dificultats diverses de comunicació.

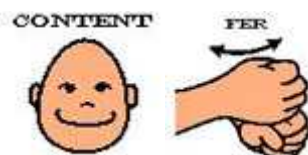


Figura 3.1.1 – Icones SPC

Aquest sistema utilitza imatges en lloc de símbols per expressar les necessitats i els estats de les persones. Majoritàriament són utilitzats pels alumnes que tenen una discapacitat greu que els impedeix produir parla de forma intel·ligible o utilitzar la llengua de signes. També es poden fer servir en situacions on la comunicació oral o signada és molt difícil o gairebé impossible com pot ser en el treball amb persones que pateixen trastorns autistes, malalts intubats, o persones estrangeres amb qui és molt difícil la comunicació.

S'utilitzen en plafons de paper i comunicadors dinàmics com a un sistema de comunicació alternativa. Les imatges s'acompanyen de la paraula escrita, per facilitar-ne la comprensió, i a vegades d'un codi de color que indica la seva categoria gramatical.

Aquest sistema també esdevé un recurs pels alumnes que encara que puguin expressar-se oralment, presenten dificultats en la lecto-escritura, l'estructuració del llenguatge, la creació de frases senzilles, tenen mancances de vocabulari, dificultats d'expressió oral.... Les icones esdevenen un suport molt important que ajuda a millorar tots aquests aspectes comunicatius i d'expressió.

3.2. Continuació d'un projecte

El projecte Messenger Visual és una iniciativa duta a terme en més d'una ocasió que mai ha acabat desembocant en un programa sòlid però que s'ha anat treballant fins al punt d'aconseguir una aplicació inicial suficientment ferma com per continuar a partir d'ella. És el cas, doncs, del projecte que es continuarà.

L'aplicació inicial comprèn l'estructura bàsica d'un aplicatiu d'aquest tipus, utilitzant els patrons de capes i, així, independitzant la part de presentació amb la part de comunicació i la part de persistència.

La part de comunicació desenvolupada està basada en sockets i és la més adient en un sistema com el que es planteja. S'estructura de manera que hi ha un servidor central que rep peticions i clients que es connecten a ell.

La interfície gràfica desenvolupa un paper únicament de prova del sistema de comunicació per validar que aquest funciona correctament, però no respecta els requeriments dels usuaris de manera que només es pot prendre com a model.

La pantalla de Login disposa d'una entrada de nom d'usuari per teclat. Aquest nom d'usuari s'envia al servidor central, el registra a la seva llista d'usuaris connectats i retorna al client aquesta mateixa llista per tal que l'usuari disposi dels contactes en línia.



Figura 3.2.1 – Pantalla de Login

A partir d'aquí, l'usuari pot establir una conversa amb qualsevol usuari en línia seleccionant-lo de la llista. El servidor central rebrà la petició d'inici de conversa del client i l'enviarà a l'usuari destí. En aquest punt s'estableix una connexió entre clients per tal que el servidor no es sobrecarregui.



Figura 3.2.2 – Pantalla de Contactes

La pantalla de conversa disposa de dues parts: una de comunicació a través de text i una altra a través d'ícones. La part de text funciona correctament però la part de comunicació a través d'ícones només disposa de quatre ícones que només permeten la validació del funcionament del model de comunicacions.



Figura 3.2.3 – Pantalla de Conversa

La part d'administració, és a dir, el servidor, només disposa d'un botó d'encès i un botó d'apagar. Això implica que aquest és capaç de processar peticions dels usuaris client però no disposa d'un sistema de gestió d'ícones, categories, perfils i, inclús, usuaris.



Figura 3.2.4 – Pantalla de Contactes

No disposa de base de dades necessària per definir els perfils d'usuari de manera que tots els usuaris tenen els mateixos privilegis de cara a la personalització de perfils. D'aquesta manera, tots els clients disposen de la totalitat d'icones disponibles i poden conversar amb qualsevol usuari que estigui en línia.

Un sistema amb base de dades permetria la personalització de perfils i, d'aquesta manera, el fet d'utilitzar uns icones o uns altres. A més a més, per exemple, permetria definir relacions entre usuaris, ja sigui de parentesc o d'amistat i, així, realitzar llistes personals de contactes.

4. Objectius

Amb aquest escenari, la proposta de la continuació del Messenger Visual pretén aconseguir una primera versió de l'aplicació que es pugui, si més no, testejar amb un grup reduït d'usuaris per tal de definir millor els requeriments d'usuari de la funcionalitat bàsica de l'aplicació.

A partir d'aquí, cal definir subobjectius per tal de realitzar millores i donar suport a noves funcionalitats sobre l'aplicació existent.

L'objectiu principal inclou els següents subobjectius:

- Dissenyar i desenvolupar una base de dades sòlida que compregui totes les necessitats inicials del sistema així com possibles millores que es puguin realitzar en una segona versió.
- Desenvolupar un aplicatiu que insereixi la informació desestructurada d'usuaris i símbols ja existents necessaris per gestionar l'aplicació principal.
- Dissenyar un logotip per tal de representar d'alguna manera l'aplicació.
- Realitzar un estudi i desenvolupar una interfície gràfica d'usuari adaptada a les necessitats dels usuaris tenint en compte el conjunt de restriccions que aquests usuaris poden tenir així com el conjunt d'interfície d'usuari d'entrada/sortida que s'hi poden adaptar.
- Adaptar la part de comunicació intermèdia amb la interfície gràfica i la base de dades respectant, en tot moment, els patrons de disseny característics de l'aplicatiu.

5 . Desenvolupament de l'aplicació

5.1 Java i MySQL

5.1.1. Per què Java?

Java és un llenguatge orientat a objectes. Altres llenguatges com C++ també estan orientats a objectes, però són molt més impurs, ja que permeten dades definides per l'usuari que no són objectes, així com la barreja de codi orientat a objectes amb codi modular o no modular. C# es troba en la mateixa línia que Java en quant a orientació a objectes, però inclou, per eficiència, estructures de dades que no són objectes.

Independència de plataforma. Els *bytecodes* produïts al compilar un programa Java poden executar-se en qualsevol plataforma amb una màquina virtual de Java. Degut a que les xarxes contenen ordinadors de molts fabricants, la independència respecte a la plataforma constitueix una característica molt valuosa. Els llenguatges inclosos en la plataforma .NET de Microsoft també es compilen a un codi entremig independent de la plataforma, però actualment .NET no s'ha propagat a tantes plataformes com el Java.

Descàrrega dinàmica de classes. Java permet descarregar els *bytecodes* d'una classe mitjançant una connexió de xarxa, crear una instància de la classe i incorporar el nou objecte a l'aplicació en execució. En conseqüència, les aplicacions en xarxa poden distribuir-se de forma molt més senzilla (mitjançant servidors web on es guarden *bytecodes*, navegadors i *applets*) i son capaços d'incorporar objectes desconeguts en el temps de compilació. C++ no ofereix unes capacitats dinàmiques com les de Java.

Inclusió estàndard de vàries API per a la programació de xarxa. El paquet `java.net` ofereix una API d'alt nivell per treballar amb el protocol HTTP, amb les URL i els sockets. La plataforma .NET també en proporciona una, sospitosament similar a la del Java. Per exemple, qualsevol programador de Java que utilitzi .NET de seguida s'adonarà de la similitud entre les classes `System.Net.Sockets.TCPClient` i `java.net.Socket`. C++ també pot

treballar amb sockets però exigeix un major esforç al programador, la lectura del codi es més complicada i les API tradicionals no estan tant orientades a objectes com les de Java.

Java es manté al dia en quant a tendències en tecnologies distribuïdes. Encara que Microsoft va apostar per les tecnologies de serveis web abans que Sun, avui en dia Java disposa d'una sèrie d'eines excel·lents per treballar amb aquestes noves tecnologies.

Comunitat d'usuaris. Java disposa d'una comunitat molt àmplia i internacional d'usuaris preocupats per difondre el llenguatge i per perfeccionar-lo.

Java s'ensenya a moltes universitats. És el cas de l'Escola Politècnica de Mataró. Per tant, Java és el llenguatge més treballat i el més còmode per desenvolupar.

Disponibilitat d'eines gratuïtes i de codi obert. Java disposa d'eines de desenvolupament gratuïtes d'excel·lent qualitat, com és el cas del NetBeans, entorn de desenvolupament integral que s'ha utilitzat per a l'elaboració d'aquest projecte. En el cas de .NET, l'eina de desenvolupament oficial és Visual Studio .Net; una molt bona solució i completament integrat amb Windows i SQL Server. Té en contra el seu preu, uns 2500 dollars actualment i que només es pot utilitzar en entorns de treball Windows.

5.1.2. Per què MySQL?

El món de les bases de dades és immens, però es centrarà majoritàriament en les bases de dades de codi obert (Open Source). Aquestes Bases de Dades tenen la particularitat de tenir versions que estan sota llicència GPL, això fa que per a aplicacions educatives i divulgatives, el seu cost sigui mínim o nul. En aplicacions comercials, el cost de les llicències comparat amb les Bases de Dades propietàries és purament testimonial.

El MySQL no es limita a la comunitat Open Source, també és portable a sistemes operatius comercials com el Windows (NT/2000/95/98/ME/XP), Solaris i altres. Un dels èxits de la seva expansió ha estat la inclusió d'aquesta BD en quasi totes les distribucions Linux, a més d'estar disponible en quasi totes les plataformes *hardware* i sistemes operatius.

És possiblement una de les BD més ràpides que es poden trobar, a més de consumir pocs recursos de la màquina on està instal·lada.

Es tracta d'un sistema de BD relativament fàcil d'instal·lar i administrar enfront d'altres productes del mercat. Es recomana per iniciar-se en el món de les BD, ja que disposa d'infinitat d'utilitats, manuals i documentació que la immensa comunitat d'usuaris s'ha encarregat de realitzar desinteressadament.

La gestió de la base de dades utilitza l'SQL (Structured Query Language) a més d'utilitzar protocols de comunicació de bases de dades desenvolupades per Microsoft, mitjançant ODBC o ADO.

Ofereix una gran capacitat de connectivitat, ja que molts clients es poden connectar simultàniament al servidor, i poden utilitzar alhora diferents Bases de Dades. Es pot accedir de forma interactiva utilitzant diferents interfícies que permeten introduir i visualitzar les consultes. A més, disposa d'una gran varietat d'interfícies de programació per al seu accés com són C, Perl, Java, PHP, Python, etc.

Es tracta d'una Base de Dades preparada per al treball en xarxa i s'hi pot accedir des de qualsevol lloc d'Internet com a resultat de la seva gran fiabilitat en matèria de control de permisos i seguretat d'accés.

A més dels manuals i fòrums de discussió, molt participatius, el MySQL ofereix un servei de contractes de manteniment i suport per als desenvolupadors o les corporacions que necessitin un acord més formal; aquest terme és fonamental per al desenvolupament d'aplicacions empresarials.

5.2 Model de comunicacions

5.2.1 Comunicació Client – Servidor

Primera connexió del client

En primera instància cal remarcar que perquè es produeixi la comunicació client-servidor és condició indispensable el fet que la màquina que conté el servidor tingui el programa en funcionament i hagi arrancat el servidor pròpiament dit. Al arrancar el servidor, el programa entra en un bucle infinit que escolta permanentment la xarxa pel port per el qual s'ha assignat prèviament que es vol que el servidor atengui les peticions.

Un cop el servidor està pendent de les peticions entrants, ja es pot arrancar qualsevol dels clients. Els clients saben, perquè ho tenen en la seva configuració, la direcció IP on es troba la màquina que conté el servidor i per quin port està pendent de les peticions entrants. Un cop el client ja sap la identificació de l'usuari que s'ha connectat al programa, envia una petició de connexió al servidor a la IP i el port destí.

El servidor veu una petició de connexió entrant i accepta la connexió. Al acceptar la petició es crea la connexió entre les dos aplicacions i, això es tradueix en l'establiment d'un canal vinculat a les dos aplicacions i la inicialització d'un buffer contenidor de bytes que permet l'intercanvi d'informació entre les aplicacions.

Arribat aquest punt, es guarden en un contenidor les dades del client. Això significa que el servidor ja disposa de la identificació del client, la seva direcció IP i el port per el qual el servidor del client escolta peticions. Un cop el servidor disposa de tota la informació que necessita del client, aprofitar el canal obert per enviar-li la informació de tots els usuaris que estan connectats. En aquest punt, el client ja pot arrancar la seva llista de contactes i es tanca la connexió entre el client i el servidor fins a futures peticions.

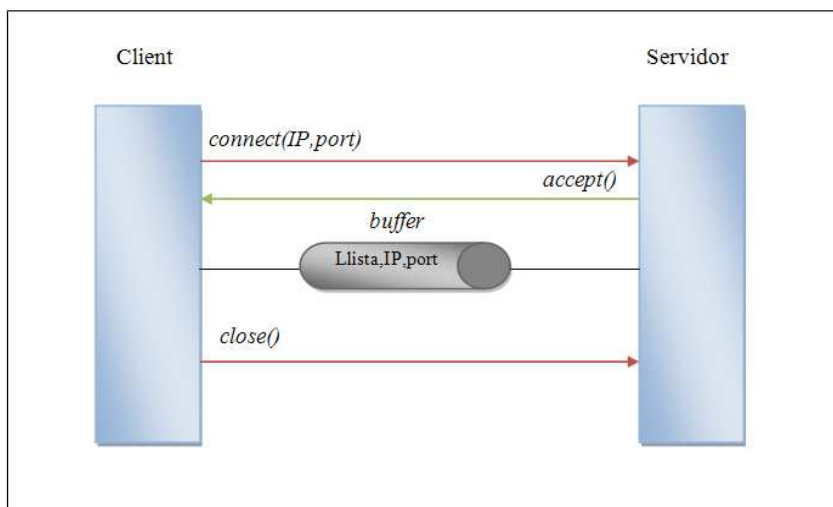


Figura 5.2.1.1 - Esquema de comunicació client-servidor

Actualització de la llista de contactes

Periòdicament els clients demanen una actualització al servidor per tal de mantenir en tot moment la llista de contactes online actualitzada. Per fer-ho, el client envia una petició de connexió al servidor. Quan aquest accepta, el client envia pel buffer la petició on demana la llista de contactes online i, el servidor, li subministra.

Un cop el client disposa de la llista actualitzada de contactes online, tanca la connexió amb el servidor. Cal remarcar el fet que el servidor en tot moment serveix les peticions seguint una política de tipus FIFO (First In First Out). Això significa que la primera petició que li arriba és la primera a la que atenció i, per tant, posa en espera les demés. Un cop atesa la primera, atenció la següent de la cua, fins la darrera.

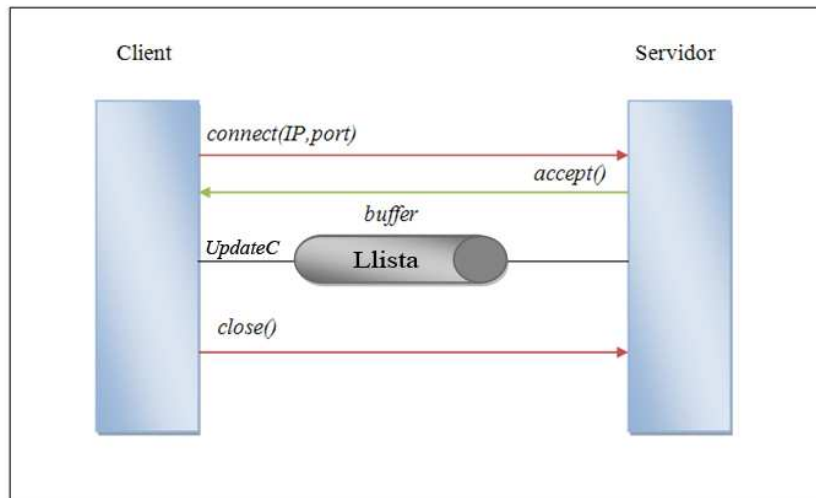


Figura 5.2.1.2 - Esquema comunicació actualitzar contactes

Petició d'informació de contacte

Quan un client vol iniciar una conversa amb un altre client ho comunica al servidor i aquest li subministra la IP i el port per el qual està escoltant aquest client, permetent així, que el servidor del client pugui establir la connexió amb l'altre client.

Per fer-ho es seguiria la metodologia del cas anterior, la única cosa que canvia és la petició que fa el client, on enlloc de demanar una actualització de contactes, demana les dades del contacte amb el que vol parlar. El servidor respon subministrant la informació demanada i, posteriorment, un cop el client rep la informació, tanca la connexió. El client, amb aquesta informació, procedeix a iniciar la conversa.

Desconnexió d'un client

Quan un client es desconnecta, segueix l'esquema vist fins ara. Primer estableix la connexió amb el servidor i a continuació envia al servidor el missatge informant que el client es desconnecta.

En aquest punt el servidor actualitza la seva llista de contactes connectats traient d'ella el client que s'ha donat de baixa i, acte seguit, actualitza a tota la seva llista de contactes enviant la llista actualitzada de contactes connectats.

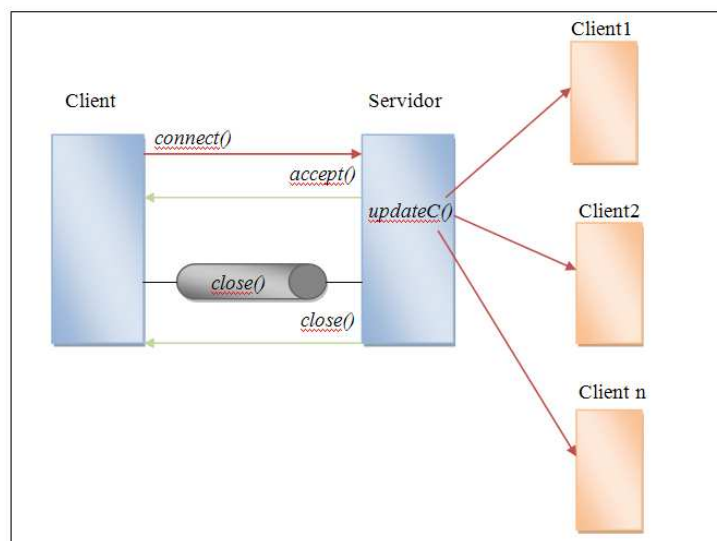


Figura 5.2.1.3 - Esquema comunicació desconnexió

5.2.2. Comunicació Client – Client

Tal i com s'ha comentat anteriorment, els clients disposen d'un petit servidor, de funcions limitades, que té menys tasques a controlar que el servidor principal, però que permet que les converses entre clients es facin independentment del servidor. El motiu d'aquesta manera de fer es demostra pel fet que carregar el servidor en excés és molt contraproductiu.

D'aquesta manera es segueix el model que fa servir el Microsoft Messenger, on es descentralitza la conversa entre clients del servidor, i es situa en el client. D'aquesta manera ens s'estalvia sobrecarregar el servidor i es garanteix que per moltes converses simultànies que pugui mantenir un client, mai s'arribarà a saturar el seu servidor.

Així doncs, els clients disposen de un petit servidor que està escoltant la xarxa per un port concret, on cada client escolta per un port diferent per evitar errors, a l'espera de qualsevol connexió entrant que pertanyi a algun contacte que vol iniciar una conversa. Quan es rep una conversa entrant, es demana al servidor que subministri la informació d'aquell contacte (IP i port destí) per tal de poder-li respondre. Un cop es té aquesta informació, es conserva fins al final de la conversa

La conversa es produeix seguint l'esquema que es mostra a continuació. Cada cop que un dels dos clients envia un missatge s'estableix la connexió, es transfereix el missatge i es tanca la connexió.

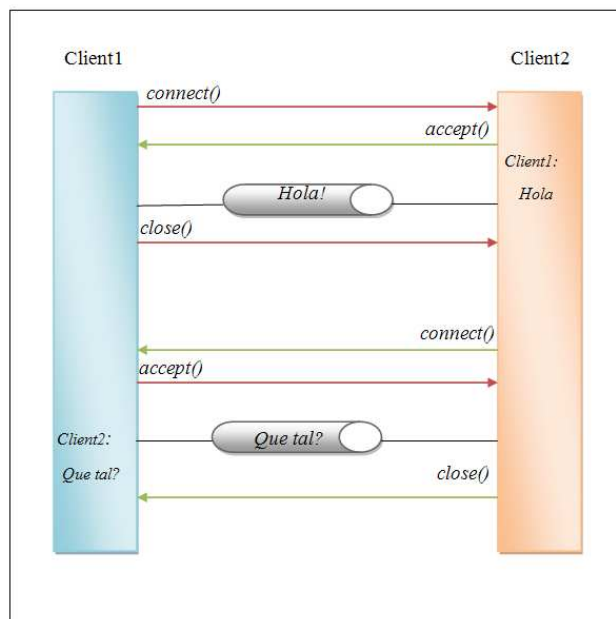


Figura 5.2.2.1 - Esquema comunicació client-client

D'aquesta manera, els servidors dels clients estaran constantment escoltant la xarxa i rebent missatges amb l'identificació de l'usuari que els envia, així sabrà en quina finestra de diàleg li pertoca col·locar-los. Sobre aquest servidor del client, cal comentar que segueix la mateixa política per servir les peticions que el servidor principal (FIFO). Cal comentar, també, que aquest servidor és totalment transparent a l'usuari, s'encén automàticament i s'executa de manera paral·lela a l'execució del client en un *thread* específic que el client ni tan sols coneix.

5.3. Model de Base de Dades

L'ús d'una base de dades és indispensable per una aplicació com és el Messenger Visual.

En el projecte anterior, com que no hi havia base de dades, qualsevol element del sistema es tractava de la mateixa manera, el que significa, per exemple, que tots els usuaris disposaven del conjunt total d'icones per establir una conversa. El fet es que els usuaris no es poden definir com un sol element ja que cadascun d'ells té un coneixement més o menys ampli de la simbologia SPC. Amb això, si els usuaris no es poden definir com un tot, caldrà establir relacions de manera que cada usuari disposi dels símbols que coneix.

En aquest punt apareixen els perfils. Els perfils no deixen de ser el concepte que vincula els usuaris amb els símbols. Així doncs, en el moment que apareix un usuari, se li assigna un perfil que contindrà, a la vegada, el conjunt d'icones disponibles per aquest perfil. Si no existís base de dades, malgrat el perfil seria un concepte sense transcendència, aquest contindria la totalitat d'icones.

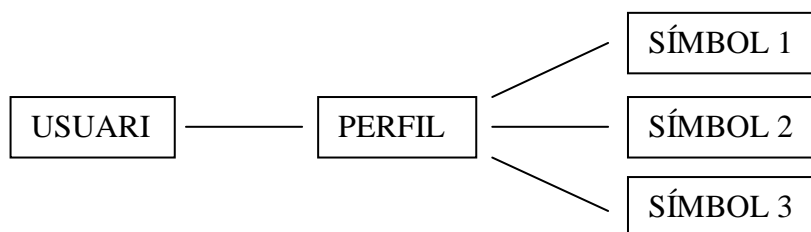


Figura 5.3.1 – Relació Usuari - Símbol

Resumint, la problemàtica principal ve donada pel fet que els usuaris que faran ús de l'aplicació no tenen el mateix coneixement de la simbologia utilitzada i, per tant, s'han d'especificar diferents perfils per a cadascun d'ells. Així doncs, un usuari té assignat un perfil i aquest defineix els símbols que ell coneix i amb els quals pot treballar. A més a més, s'estableix una connexió entre categories i la totalitat de símbols disponibles de manera que hi hagi un ordre a l'hora d'organitzar-los dins de la pròpia aplicació.

D'altra banda, un fet que mostra la necessitat d'una base de dades és en el moment de configurar llistes d'usuaris connectats per a cadascun d'ells. En aquesta versió s'ha tingut en compte que els usuaris han de disposar de llistes de contactes personalitzades ja que no tots tenen relació amb tots, ja sigui de parentesc o d'amistat. De totes maneres, malgrat consta a la base de dades, aquesta funcionalitat s'ha deixat per més endavant.

5.3.1. Model conceptual de dades

El model conceptual descriu la realitat amb un alt nivell d'abstracció de tal manera que sigui fàcil d'entendre. Així doncs:

- El concepte d'Usuari fa referència al conjunt de propietats que fan únic un usuari de l'aplicació.
- El concepte de Perfil estableix la relació entre els usuaris i els símbols que coneixen i poden utilitzar.
- El concepte de símbol inclou cadascun dels símbols específics que formen la simbologia utilitzada.
- El concepte de Categoria determina un ordre dins l'aplicació classificant les icones existents de la simbologia utilitzada de tal manera que sigui més fàcil el seu tractament.

Usuari

El concepte d'usuari ve definit pels atributs idusuari, nom, cognom1, cognom2, imatge, email i password. El nom, cognom1, cognom2, email i password tenen, tots ells, un format de dades de conjunt de caràcters encara que els últims tres són de caràcter opcional i, per tant, no són estrictament necessaris en el moment de donar d'alta un nou usuari al sistema.

L'idusuari és un codi amb format d'enter que utilitza internament l'aplicació donant la màxima eficiència possible en el moment de realitzar consultes i transaccions d'informació. Aquest codi es genera automàticament de forma incremental en el moment d'inserir un nou usuari al sistema.

La imatge té format blob i, per tant, és guardada íntegrament a la pròpia base de dades. La metodologia alternativa és assignar a aquest camp un format de conjunt de caràcters especificant la ruta per trobar la imatge al servidor. El fet de valorar l'ús d'una o altra metodologia entra dins de l'estudi del disseny i, en aquest cas, s'ha apostat per guardar les dades en format binari degut a les petites dimensions de les imatges a emmagatzemar.

De fet, els camps email i password no s'utilitzen a l'aplicació però s'inclouen pel fet que es puguin donar, en un futur pròxim, altres serveis que no s'han desenvolupat en aquesta primera versió com poden ser l'enviament de novetats a través del correu electrònic d'actualitzacions del sistema o la utilització d'algun tipus de seguretat en el moment d'accedir al sistema.

Usuari			
<u>IDUsuari</u>	<pi>	Integer	<M>
Nom		Variable characters (30)	<M>
Cognom1		Variable characters (20)	<M>
Cognom2		Variable characters (20)	
Imatge		BLOB	<M>
Email		Variable characters (50)	
Password		Variable characters (20)	
IDUsuari	<pi>		

Figura 5.3.1.1 – Entitat usuari

Perfil

El concepte de perfil ve definit pels atributs idperfil, nom, genèric i descripció. El nom i la descripció tenen un format de conjunt de caràcters i s'utilitzen per especificar el que el seu nom indica. El camp descripció és opcional i, de fet, serà únicament útil de cara a l'administrador, on s'hi podrà comentar qualsevol aspecte referent al perfil en concret.

L'idperfil, de la mateixa manera que en el cas de l'usuari, no deixa de ser un codi intern de l'aplicació que facilita la interconnexió entre elements de la base de dades.

Per últim, genèric és un camp que especifica si el perfil en qüestió és personalitzable o no a través d'un booleà. Els perfils genèrics permeten crear altres perfils a través d'ells, és a dir, permeten a l'administrador utilitzar-los com a patró per tal d'aprofitar les seves

assignacions internes existents. Un exemple podria ser el següent: existeixen els perfils genèrics adult i nen i es decideix crear un nou perfil per un nen amb un coneixement més ampli de la simbologia utilitzada que el que descriu el perfil genèric nen. Així doncs, utilitzant com a base el perfil nen, se'n pot crear un de nou amb varies modificacions respecte el genèric.

Perfil			
<u>IDPerfil</u>	<pi>	<u>Integer</u>	<M>
Nom		Variable characters (30)	<M>
Generic		Boolean	<M>
Descripcio		Text (500)	
IDPerfil	<pi>		

Figura 5.3.1.2 – Entitat perfil

Símbol

El concepte de símbol ve definit pels atributs idsímbol, nom, imatge i descripció. Els camps idsímbol, nom i descripció tenen la mateixa funció que en els conceptes anterior perfil. Per altra banda, la imatge conté la icona concreta de la simbologia utilitzada en format binari i es guarda íntegrament a la base de dades.

Símbol			
<u>IDSímbol</u>	<pi>	<u>Integer</u>	<M>
Nom		Variable characters (30)	<M>
Imatge		BLOB	<M>
Descripcio		Text (500)	
IDSímbol	<pi>		

Figura 5.3.1.3 – Entitat símbol

Categoria

Pel que fa al concepte de categoria, aquesta ve definida pels atributs idcategoria, nom, imatge i descripció. Els camps idcategoria, nom i descripció funcionen de la mateixa manera que en els casos anteriors, en canvi, però, el camp imatge fa referència a la icona específica que simbolitza el conjunt de símbols que conté.

Categoria			
<u>IDCategoria</u>	<pi>	Integer	<M>
Nom		Variable characters (30)	<M>
Imatge		BLOB	<M>
Descripcio		Text (500)	
IDCategoria <pi>			

Figura 5.3.1.4 – Entitat categoria

El model conceptual defineix les relacions entre els conceptes d'usuari, perfil, categoria i símbol. Els usuaris tenen assignat un perfil que especifica quins símbols pot utilitzar. D'aquesta manera, un símbol pot formar part de més d'un perfil i un perfil pot estar assignat a més d'un usuari. Per altra banda, una símbol és gestionat per una categoria en concret.

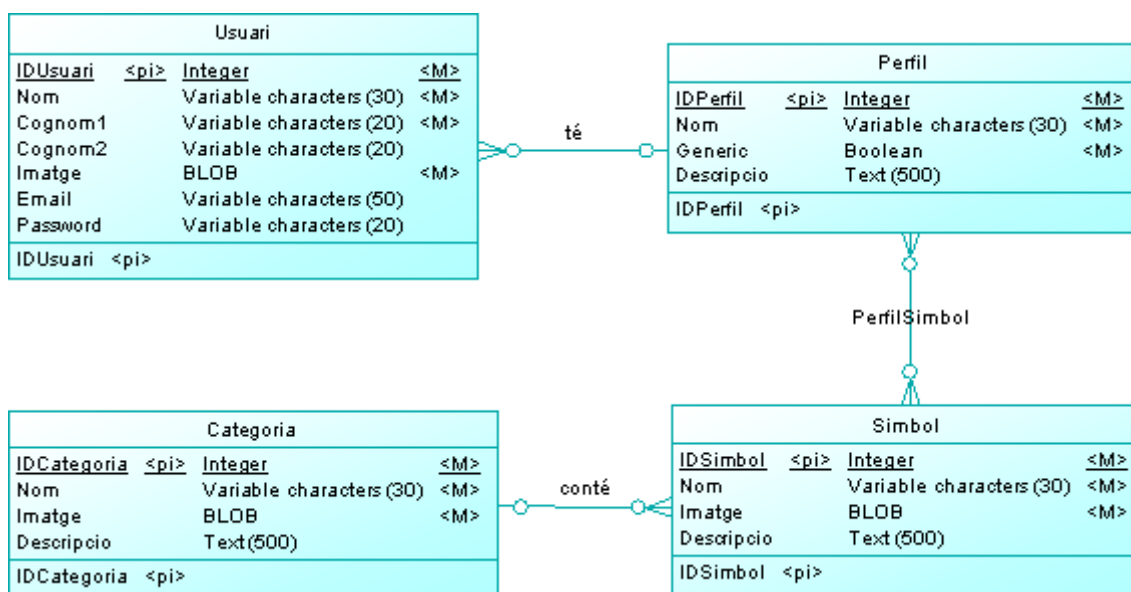


Fig. 5.3.1.5 - Model conceptual de dades

5.3.2. Model relacional de dades

El model relacional estableix una correspondència senzilla entre la descripció de les dades del model conceptual amb l'estructura física de la base de dades. Així doncs, aquest model conceptual mostra les taules i els vincles definitius que assoleix la base de dades tot mostrant de forma esquemàtica les seves relacions.

Els canvis més substancials respecte el model conceptual són:

- L'assignació de formats de dades vàlids pel sistema gestor de bases de dades utilitzat, en aquest cas, MySQL.
- La incorporació de les claus primàries dels conceptes perfil i categoria com a claus foranes dels conceptes usuari i símbol, respectivament. Això s'explica pel fet que un usuari només pot tenir assignat un perfil i un símbol només pot ser d'una categoria en concret.
- L'aparició d'una nova taula anomenada perfilsímbol. Aquesta taula emmagatzema les relacions entre perfil i símbol i es deu al fet que un perfil pot tenir diferents símbols assignats i, a la vegada, un símbol pot pertànyer a més d'un perfil.

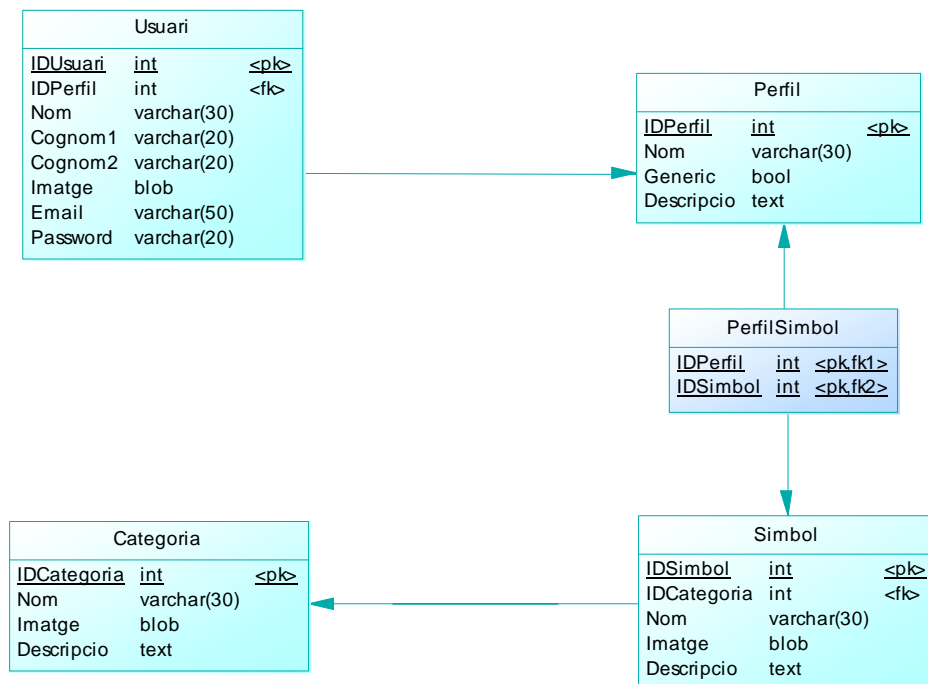


Fig. 5.3.2.1 - Model relacional de dades

5.4. Interfície gràfica

La interfície gràfica és la part més important de l'aplicació i necessita d'un estudi previ molt acurat per tal que aquesta s'ajusti perfectament a les necessitats i possibilitats de l'usuari. Els aspectes més importants pels que s'han de vetllar són la senzillesa i la comprensió de l'aplicació de tal manera que no sigui necessària ajuda externa per fer ús d'ella.

S'ha tingut en compte el fet que la interacció per part de l'usuari amb l'aplicació serà en la seva totalitat a través del *mouse* i/o a través de commutadors. Així doncs, la interacció amb l'ordinador a través de teclat es troba únicament en el moment de donar d'alta un usuari i en les operacions de manteniment de l'aplicació per part de l'administrador.

5.4.1. Pantalla de benvinguda

La pantalla de Login del projecte anterior disposava de l'entrada per teclat d'un nom d'usuari i, amb la posterior acceptació, s'accedia al programa. El problema principal i que genera una contradicció és el fet de plantejar-se escriure un nom d'usuari quan justament el programa va dirigit, en part, a persones amb problemes de llenguatge.



Figura 5.4.1.1 – Pantalla de Login

Així doncs, la metodologia d'actuació s'ha canviat i és la següent:

Un cop s'accedeix a l'aplicació, apareix la pantalla de benvinguda. Aquesta permet a l'usuari entrar a l'aplicació amb una identificació vàlida amb la qual els altres usuaris el puguin identificar. Aquesta identificació correspondrà amb el nom introduït en el moment de donar d'alta a l'usuari i hauria de coincidir amb el seu nom real o un altre amb el qual els altres usuaris el puguin relacionar.

Així doncs, la metodologia d'actuació és senzilla i fàcilment comprensible: l'usuari selecciona amb el *mouse* o el commutador la seva fotografia del panell on apareixen totes les fotografies dels usuaris registrats i confirma la seva decisió pressionant sobre la fotografia que apareixerà ressaltada a sobre del panell. A més a més, aquest panell disposa d'un cursor de desplaçament, per tal de desplaçar cap a la dreta i cap a l'esquerra el conjunt de fotografies registrades, útil en el cas que sigui necessari si no apareix en primera instància la fotografia de l'usuari que vol accedir a l'aplicació.

Aquesta primera pantalla disposa a la part inferior d'un botó addicional anomenat "Crear compte" que porta directament a la pantalla per donar d'alta un nou usuari.

La següent figura mostra el prototip dissenyat de pantalla de benvinguda amb possibles usuaris de l'aplicació on es pot apreciar la selecció d'un usuari del panell de fotografies d'usuaris, en aquest cas "Maria Cruz", i apareix el seu nom i la seva fotografia ressaltada a sobre del panell. En aquest punt, prement sobre la fotografia de l'usuari seleccionat, l'aplicació l'inclou a la llista d'usuaris connectats i apareix la pantalla de contactes connectats per tal d'escollir un altre usuari en línia i així iniciar una conversa.

L'estudi del disseny de la pantalla comporta el fet de decidir en temes de seguretat. Com es pot observar, la pantalla actual no disposa de ningun tipus de mecanisme per mantenir la privacitat de dades però és el mètode que s'ha decidit en un primer moment ja que endinsar-se en contrasenyes portaria al mateix lloc.



Fig. 5.4.1.2 – Pantalla de Login

5.4.2. Pantalla de contactes

La pantalla de contactes apareix un cop l'usuari ha accedit a l'aplicació. Aquesta pantalla mostra la totalitat d'usuaris connectats a la xarxa i permet seleccionar-los per tal d'iniciar una conversa amb ells.

La versió del projecte anterior tenia un problema semblant al mencionat anteriorment i és que la selecció de l'usuari s'havia de fer llegint el nom de l'usuari amb el que es volia mantenir una conversa. Aquest fet plantejava altra vegada el dilema de què fer en aquests casos. No obstant això, en aquest cas, la solució passa per afegir al nom de l'usuari, la fotografia que l'identifica i així, es trenquen les barreres dels problemes de llenguatge.



Figura 5.4.2.1 – Pantalla de Contactes

Utilitzar o no fotografies ja defineix que el sistema s'ha d'encarar cap a bases de dades ja que com a mínim s'ha d'establir un vincle entre els noms dels usuaris i les seves fotografies.

La metodologia d'actuació sobre la nova pantalla de contactes és la següent:

La selecció de l'usuari amb el qual es vol iniciar una conversa es duu a terme a través del mouse i/o commutador. A més a més, de la mateixa manera que en el cas de la pantalla de benvinguda, aquesta pantalla disposa de dos cursors de desplaçament, cap amunt i cap avall, que permeten desplaçar la llista d'usuaris per tal de poder-ne seleccionar un que en primera instància no apareix en ella.

L'actualització dels contactes es duu a terme de forma automàtica i, per tant, aniran apareixen a la llista els contactes que vagin entrant a l'aplicació, i s'aniran esborrant els que es vagin desconnectant.

En aquesta primera versió, tots els usuaris formen part d'un mateix grup i, per tant, no s'ha estudiat el fet de que cada usuari disposi d'una llista d'usuaris personalitzada. Així doncs, malgrat dos usuaris no es coneguin, apareixeran connectats a la llista d'altre.

La següent figura mostra el prototip dissenyat de pantalla de contactes connectats amb possibles usuaris de l'aplicació.



Fig. 5.4.2.2 – Pantalla de Contactes

5.4.3. Pantalla de conversa

La pantalla de conversa és la pantalla que apareix en el moment de seleccionar un usuari de la llista d'usuaris connectats i dóna totes les funcionalitats per establir una conversa utilitzant la simbologia utilitzada.

La pantalla de conversa del projecte anterior contemplava poder-se comunicar utilitzant tant text com la simbologia SPC. L'estructuració de la pantalla no es pot analitzar ja que és només una versió de prova i, per tant, no s'ha pensat en el disseny que hauria de tenir.



Figura 5.4.3.1 – Pantalla de Conversa

En la nova versió, la pantalla es distribueix en cinc parts:

- A la part superior de la pantalla es troben el conjunt de categories que gestionen l'ordre dins la pròpia aplicació. Si es vol, doncs, accedir als símbols, per exemple, d'accions o calendari, s'ha de moure el mouse o, en el seu defecte, el commutador per desplaçar-se fins a la icona en qüestió. A més a més, aquesta barra superior disposa de dos cursors de desplaçament, cap a la dreta i cap a l'esquerra, per tal de poder seleccionar una categoria que en un primer moment no apareix en pantalla.
- La barra situada a l'esquerra són un conjunt d'icones d'ús freqüent que faciliten a l'usuari mantenir converses de forma fluida ja que deixen a mà les icones més bàsiques i més utilitzades.
- La barra situada a la dreta conté el conjunt de símbols que formen part de la categoria seleccionada del menú superior. Així doncs, si a la barra superior es troba seleccionada la icona d'accions, els símbols que apareixeran a aquesta barra seran del tipus anar, ballar, cantar, etc. A més a més, aquesta barra disposa de cursors de desplaçament, cap a dreta i esquerra, que permeten a l'usuari desplaçar el conjunt de símbols a la vista per tal que se'n puguin seleccionar d'altres que formen part de la categoria seleccionada però que no apareixen en aquell moment a la llista actual.

- El quadre de text situat a la part inferior de la pantalla permet a l'usuari crear una llista de símbols, en forma de frase, pregunta, etc., per tal d'enviar-los seguidament a l'usuari. Aquest seguit de símbols es crearan a partir de la selecció d'icones dels menús laterals tot utilitzant el menú superior per canviar els símbols mostrats a la barra de
- símbols situada a la dreta. En el moment en què es vulguin enviar el conjunt de símbols a l'altre usuari, només cal prémer sobre el botó situat a la dreta del quadre de text per tal de fer efectiu el seu enviament. Un cop fet, el quadre borra els símbols enviats i permet crear una nova llista de símbols per ser enviada.
- La part central de la pantalla mostra la conversa entre els dos usuaris. Cadascuna de les línies de text està formada per la fotografia de l'usuari que es comunica seguit dels símbols que componen la frase enviada. D'aquesta manera, el primer usuari en aparèixer a l'àrea de text és l'usuari que inicia la conversa i, a partir d'aquí, la seqüència de missatges depèn de l'evolució de la conversa.



Fig. 5.4.3.2 – Pantalla de Conversa

6. Limitacions

L'aplicació presenta tres limitacions principals:

- Els usuaris no presenten limitacions de conversa entre ells malgrat poden tenir diferents perfils assignats i, per tant, és possible que no tinguin un mateix coneixement de la simbologia utilitzada. Aquesta limitació presenta la qüestió de què s'han de fer amb els símbols rebuts que no incorpora el perfil de l'usuari receptor. Les dues possibles solucions passen per mostrar la icona tal i com envia l'emissor del missatge o convertir la icona en text i, així, rebre al receptor part d'icones i part de text en una mateix missatge.
- La gestió dels perfils personalitzats els duu a terme l'administrador de l'aplicació. En el moment de donar d'alta un usuari, l'assignació del perfil el durà a terme la persona que insereixi les dades del nou usuari però, aquest, només tindrà disponibles perfils d'estil genèric. En aquest punt, l'administrador de l'aplicació ha de gestionar la personalització del perfil genèric segons els coneixements de l'usuari en qüestió. Aquesta limitació crea el debat de si la mateixa persona que dóna d'alta a un usuari pot gestionar la personalització de perfils genèrics o és únicament gestionable per part de l'administrador de l'aplicació.
- Els usuaris no disposen de llistes d'usuaris personals i, per tant, tots els usuaris donats d'alta a l'aplicació disposaran de la mateixa llista de contactes connectats sense tenir en compte si els usuaris tenen alguna relació de parentesc o simplement es coneixen. Aquesta limitació mostra una possible millora de l'aplicació malgrat s'hagi d'estudiar la metodologia amb la qual es durà a terme tenint en compte les limitacions que presenten tant els usuaris com l'administrador d'aquesta aplicació, depenent de la solució adoptada.

7. Continuitat del projecte

La continuïtat del projecte suposarà el test de les funcionalitats dutes a terme fins ara, l'estudi de la millora d'aquestes funcionalitats actuals i la incorporació de noves funcionalitats a l'aplicació.

7.1. Test de l'aplicació

Un primer pas que es podria considerar en una següent iteració seria el fet de realitzar una petita prova a partir del punt en que es troba actualment l'aplicació desenvolupada. És un bon moment per veure si la direcció, bàsicament del disseny i la funcionalitat de la interfície gràfica d'usuari, segueix un camí amb possibilitats o realment l'estudi previ realitzat no acaba d'ajustar-se a les verdaderes necessitats dels usuaris. És probable que s'hagin de tenir en compte aspectes més enllà dels estudiats i, per tant, per no complicar el fet de fer i desfer un cop el projecte es trobi molt avançat, es podria prendre aquest punt com a punt de partida. Cal tenir en compte que el disseny no l'han dut a terme educadors socials sinó estudiants i que, per tant, és comprensible que hi hagi errades que es troben fora dels límits del coneixement d'ells i que s'han solucionat, no com a professionals, sinó amb lògica.

7.2. Millora de funcionalitats

Malgrat l'aplicació desenvolupada compleix la funcionalitat bàsica del servei, hi ha alguns aspectes millorables i d'altres que no tenen resposta i necessiten d'un estudi més acurat o la intervenció d'educadors socials que donguin el seu punt de vista de la situació.

Com a aspectes més importants i que, per altra banda, de moment no tenen resposta són les limitacions que ofereix la aplicació en els apartats de seguretat, personalització de perfils i interacció entre usuaris amb diferents nivells de coneixement de la simbologia utilitzada.

Pel que fa a la seguretat, cal fer un estudi de com els usuaris poden accedir a les seves dades de manera que es tinguin en compte les seves limitacions alhora del fet de no vulnerar la privacitat de les seves dades. Actualment, l'usuari accedeix al seu perfil prement sobre la seva fotografia en el moment d'entrar a l'aplicació. Aquesta solució no té en compte cap tipus de privacitat de dades ja que qualsevol usuari pot accedir a qualsevol perfil de qualsevol altre usuari però és la solució més còmode. Qualsevol solució d'entrada d'una contrasenya per teclat crea el debat de si realment no es pot fer d'una altra manera.

Com a aspecte secundari, cal ajustar la interfície gràfica de tal manera que sigui funcional per a qualsevol resolució de pantalla. Actualment, aquesta es troba limitada en unes dimensions concretes i, per tant, molt restringida en el moment que es vol augmentar o reduir la dimensió de qualsevol de les pantalles. Seria bo poder reestructurar la pantalla i el tamany de les icones tant a pantalla completa com a diferents resolucions.

7.3. Incorporació de noves funcionalitats

En quan a noves funcionalitats, destaca tota la part d'administrador, pel que fa al fet d'inserir, modificar i esborrar categories, símbols i perfils d'usuari.

Les següents pantalles poden ser possibles pantalles d'inserir, modificar i esborrar categories, símbols i perfils d'usuari. No és necessari explicar-les ja que es sobreentenen perfectament.

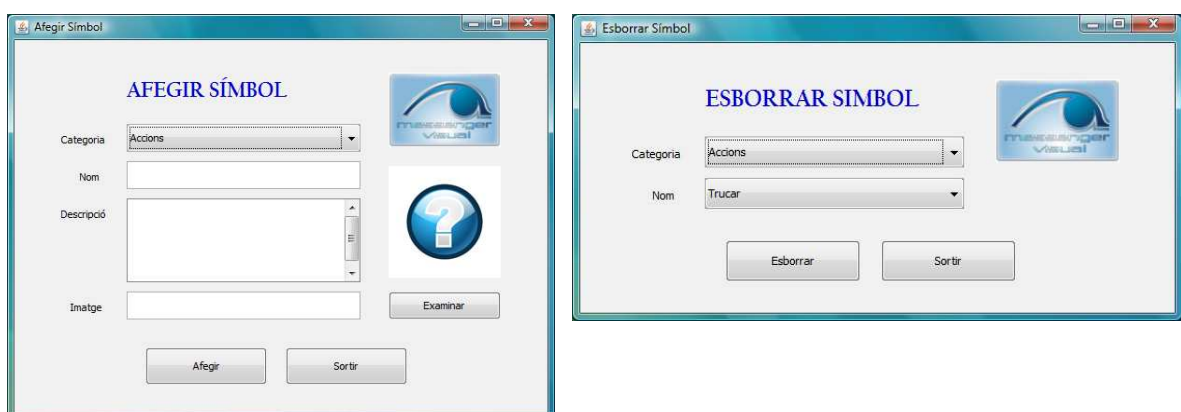


Figura 7.3.1 – Pantalles d'afegir i esborrar símbol



Figura 7.3.2 – Pantalla de modificar símbol

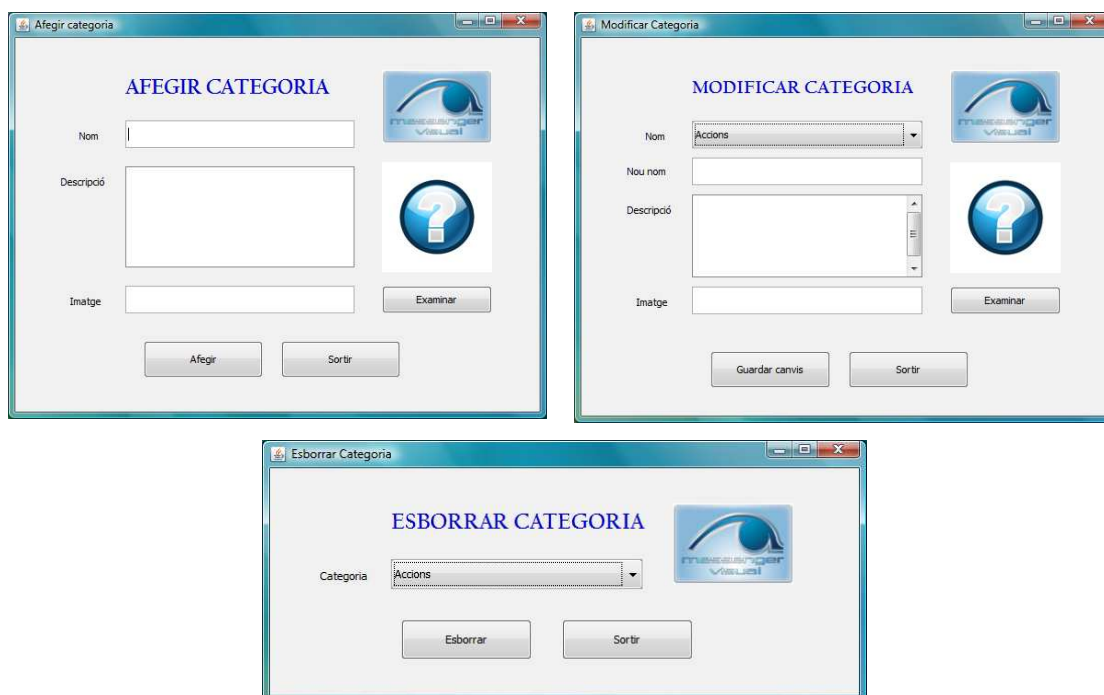


Figura 7.3.3 – Pantalles d'afegir, modificar i esborrar categoria



Figura 7.3.4 – Pantalla d'esborrar perfil

El que són les pantalles d'afegir i modificar perfil s'han d'estudiar ja que per crear o modificar perfils s'hi han d'afegir i esborrar símbols. Per tant, una possible solució passaria per elaborar una tipus de llista on es poguessin seleccionar els símbols que pertanyen al perfil que s'està afegint o modificant. De totes maneres, cal veure com estructurar la pantalla per tal que sigui fàcil d'utilitzar tenint en compte el gran nombre de símbols que es disposa.

Per altra banda, la personalització dels perfils per a cadascun dels usuaris també és motiu de discussió. Quan un usuari es dona d'alta al sistema, se suposa que aquest tràmit és dut a terme per una persona de confiança que coneix a l'usuari final de l'aplicació, però que probablement no conegui al 100% el nivell de coneixement de l'usuari de la simbologia utilitzada. En aquest moment es suposa que aquesta persona li assignarà un perfil genèric a l'usuari però que més tard, l'administrador de l'aplicació ajustarà i, per tant, personalitzarà el perfil per l'usuari en concret.

Per últim, una nova funcionalitat ben últim consistiria en crear llistes d'usuaris per un usuari en concret. Els usuaris no disposen de llistes d'usuaris personals i, per tant, tots els usuaris donats d'alta a l'aplicació disposaran de la mateixa llista de contactes connectats sense tenir en compte si els usuaris tenen alguna relació de parentesc o simplement es coneixen. Aquesta limitació mostra una possible millora de l'aplicació malgrat s'hagi d'estudiar la metodologia amb la qual es durà a terme tenint en compte les limitacions que presenten tant els usuaris com l'administrador d'aquesta aplicació, depenent de la solució adoptada.

8. Conclusions

El Messenger Visual és un innovador servei de missatgeria instantània basat en icones amb l'objectiu de millorar les comunicacions entre persones discapacitades. A través dels resultats obtinguts en un estudi previ, es pot dir que la combinació de la comunicació visual amb les noves tecnologies pot ser de gran utilitat per a persones amb problemes de llenguatge. A més a més, el programa es pot adaptar fàcilment per a persones amb mobilitat reduïda adaptant la interfície d'usuari. Cal dir que la avaluació del software en procés de desenvolupament és una bona oportunitat per definir els requeriments d'usuari en el disseny ja que apareixen qüestions que probablement no trobin la solució òptima fins que l'aplicatiu es provi en un laboratori.

Per exemple, una primera pregunta que va sorgir durant l'estudi previ és què passa quan un usuari avançat, per exemple, un adult amb un vocabulari més ampli, prova de dialogar amb un usuari novell, per exemple, un nen. La opció més senzilla és mostrar la icona. Un altre mecanisme és que els dos clients acordin prèviament el conjunt d'icones disponibles depenent del perfil d'usuari, on un usuari avançat no pot utilitzar un símbol que l'altre usuari no entén o encara no coneix. La última alternativa considerada és enviar nous símbols progressivament, de manera que es tracti d'un mètode per anar incrementant el nombre d'icones conegudes.

Per acabar, una altra problema que cal estudiar de cara a la versió final de l'aplicatiu és la gestió de perfils, per exemple, com controlar els perfils sense utilitzar contrasenyes amb text, i temes de seguretat, per exemple, la privacitat de les dades personals en una comunicació a través d'Internet.

Annex I – Glossari de termes

BLOB

Els BLOB (Binary Large Object, grans objectes binaris) són elements utilitzats en les bases de dades per emmagatzemar dades de gran tamany que canvien de forma dinàmica. No tots els Sistemes Gestors de Bases de dades són compatibles amb BLOB.

Generalment, aquestes dades són imatges, arxius de so i altres objectes multimèdia; a vegades s'emmagatzemen com a BLOB codis de binaris.

El terme blob es referia originalment a trossos amorfs de codi, i va ser inventat per Jim Stakey. Amb el temps, Terry McKiever, un encarregat de mercatècnia, va idear un acrònim: Basic Large Object (gran objecte bàsic). Però va ser Informix qui va idear l'actual acrònim per a BLOB.

El tipus de dades i la seva definició es van introduir per representar dades que anteriorment no estaven definits en les bases de dades per a computadors, però que van fer possible abaratir els discs d'emmagatzematge.

Bytecode

Un bytecode és un codi binari intermedi més abstracte que el codi màquina. Habitualment és tractat com un fitxer binari que conté un programa executable similar a un mòdul objecte, que és un fitxer binari produït pel compilador on el seu contingut és el codi objecte o codi màquina.

Com a codi intermedi, es tracta d'una forma de sortida utilitzada pels implementadors de llenguatges per reduir la dependència respecte del hardware específic i facilitar la interpretació. Menys freqüentment s'utilitza el bytecode com a codi intermedi. Alguns sistemes, anomenats traductors dinàmics o compiladors just-in-time tradueixen el bytecode

a codi màquina immediatament abans de la seva execució per a millorar la velocitat d'execució.

Els programes en bytecode solen ser interpretats per un intèrpret de bytecodes (en general anomenat màquina virtual, donat que és anàleg a l'ordinador). El seu avantatge és la seva portabilitat: el mateix codi binari pot ser executat en diferents plataformes i arquitectures. És el mateix avantatge que presenten els llenguatges interpretats. No obstant, com que el bytecode és en general menys abstracte, més compacte i més orientat a la màquina que un programa pensat per la seva modificació per humans, el seu rendiment sol ser millor que el dels llenguatges interpretats.

Thread

La màquina virtual de Java (JVM) és un sistema multi-thread, és a dir, és capaç d'executar diverses seqüències d'execució (programes) simultàniament. La JVM gestiona tots els detalls, assignació de temps d'execució, prioritats, etc., de forma similar a com gestiona un Sistema Operatiu múltiples processos.

La diferència bàsica entre un procés de Sistema Operatiu i un Thread Java és que els Threads corren dins de la JVM, que és un procés del Sistema Operatiu i per tant comparteixen recursos, inclosa la memòria i les variables i objectes allà definits. A aquest tipus de processos on es comparteixen els recursos se'ls anomena de vegades "processos lleugers" (lightweight process).

Annex II – Script d’inserció inicial a la BBDD

La simbologia distribuïda per part de la Fundació El Maresme Pro Persones Amb Disminució Psíquica consisteix en icones en format JPG amb el nom que correspon a l'icona i organitzades en carpetes segons la categoria a la que pertanyen. Cal tenir en compte que la inserció de les icones a la base de dades pot convertir-se en una feina feixuga per la quantitat total d'icones que disposa la simbologia utilitzada. Així doncs, s'ha optat per crear un script que insereix automàticament les icones a la base de dades de manera que es troben llestes per ser tractades.

Per executar correctament l'script, cal crear una carpeta a C:/ que s'anomeni Imatges i a dins crear una carpeta pels símbols, que s'anomeni Símbols, i una pels usuaris, que s'anomeni Usuaris. A dins de la carpeta de símbols s'hi afegiran tots els símbols estructurats en carpetes de categories i a dins de la carpeta d'usuaris, la totalitat d'usuaris.

Pel que fa als usuaris, L'script crea una entrada per cada usuari dins de la base de dades assignant-li un número d'identificació d'usuari, un nom i cognom i la fotografia. Altres possibles camps com el perfil, el correu electrònic i el password s'han d'assignar posteriorment.

Pel que fa a les categories, l'script crea una entrada per cada categoria dins de la base de dades assignant-li un número d'identificació de categoria i un nom. Els camps descripció i imatge s'han d'assignar manualment.

Pel que fa als símbols, l'script crea una entrada per cada símbol dins de la base de dades assignant-li un número d'identificació de símbol, un número d'identificació de categoria, un nom i la imatge del símbol. El camp descripció s'ha d'inserir manualment en el cas que a l'administrador li sigui necessari.

El codi utilitzat alhora de desenvolupar l'script és el següent:

```
package script;

import java.io.*;
import java.sql.*;
import javax.swing.JOptionPane;

/**
 * @author Xavier Cruz
 */
public class Main {

    private static Connection conn;
    private static final String DRIVER = "com.mysql.jdbc.Driver";
    static String bd = "messenger_visual";
    static String login = "root";
    static String password = "";
    static String url = "jdbc:mysql://localhost:3306/" + bd;

    public static void main(String[] args) throws Exception {

        inserirSimbols();
        inserirUsuaris();
    }

    public static void inserirSimbols() {

        try {
            connectar();

            PreparedStatement stmt;
            InputStream is;

            File directori = new File("C:\\\\Imatges\\\\Simbols");
            File[] categories = directori.listFiles();
            for (int i = 0; i < categories.length; i++) {
```



```

        File[] icones = categories[i].listFiles();
        String lletrescat = categories[i].toString().substring(10);

        stmt = conn.prepareStatement("INSERT INTO
Categoria(IDCategoria,Nom) VALUES (?,?)");
        stmt.setInt(1, i + 1);
        stmt.setString(2, lletrescat);
        stmt.executeUpdate();

        int nlletrescat = lletrescat.length();
        for (int j = 0; j < icones.length; j++) {

String icona = icones[j].toString().substring(11 + nlletrescat);
        System.out.println(icona);
        if (!icona.contains("Thumbs.db")) {
            stmt = conn.prepareStatement("INSERT INTO
Simbol(IDCategoria,Nom,Imatge) VALUES (?,?,?)");
            is = new BufferedInputStream(new FileInputStream(icones[j]));

            stmt.setInt(1, i + 1);
            icona = icona.substring(0, icona.length() - 4);
            stmt.setString(2, icona);
            stmt.setBlob(3, is, (int) icones[j].length());
            stmt.executeUpdate();
        }
    }
} catch (Exception e) {
    new Exception(e);
}
}

public static void inserirUsuaris() {

    try {
        connectar();
    }
}

```

```
PreparedStatement stmt;
InputStream is;

File directori = new File("C:\\Imatges\\Usuaris");
File[] usuaris = directori.listFiles();
for (int i = 0; i < usuaris.length; i++) {

    String icona = usuaris[i].toString().substring(11);
    System.out.println(icona);
    if (!icona.contains("Thumbs.db")) {
        stmt = conn.prepareStatement("INSERT INTO
Usuari(IDUsuari,Nom,Imatge) VALUES (?,?,?)");
        is = new BufferedInputStream(new FileInputStream(usuaris[i]));
        stmt.setInt(1, i + 1);
        icona = icona.substring(0, icona.length() - 4);
        stmt.setString(2, icona);
        stmt.setBlob(3, is, (int) usuaris[i].length());
        stmt.executeUpdate();
    }
}
} catch (Exception e) {
    new Exception(e);
}
}
```

```
public static void connectar() throws Exception {
    try {

        Class.forName(DRIVER).newInstance();
        conn = DriverManager.getConnection(url, login, password);
        if (conn != null) {
System.out.println("Conexión a base de datos " + url + " ... Ok");
        }

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, " ERROR: no hi ha
connexió amb la BD ", "Avís", JOptionPane.ERROR_MESSAGE);
    } catch (ClassNotFoundException ex) {
        System.out.println(ex);
    }
}
}
```


Annex III – Redimensionament d'imatges

Les icones utilitzades a l'aplicació necessiten unes dimensions concretes i específiques per tal que aquesta tingui un format visual comprensible. Així doncs, cal redimensionar cadascuna de les imatges que intervinguin a l'aplicació podent violar la restricció de proporcions de tamany d'imatge, opció que no permeten la majoria de programes. No sembla un problema greu però si es té en compte el volum d'imatges que intervenen a l'aplicació (entre 4000 i 5000), s'ha de localitzar una aplicatiu que automatitzi aquesta redimensió.

L'aplicatiu utilitzat amb el qual es soluciona aquesta qüestió és l'*Adobe Photoshop CS4*. Aquest programa d'edició fotogràfic permet grabar i reproduir els moviments duts a terme dins de l'aplicació entre dos instants de temps i, per tant, automatitzar qualsevol procés que s'hagi fet amb ell.

El primer pas és obrir l'arxiu a redimensionar amb el programa en qüestió. Un cop obert, a la barra d'eines superior hi apareix l'eina *Finestra* i, a dins, l'eina *Accions*. En aquest moment apareix una finestra amb l'eina *Accions* que s'utilitzarà per grabar l'acció que, posteriorment, es vol reproduir.

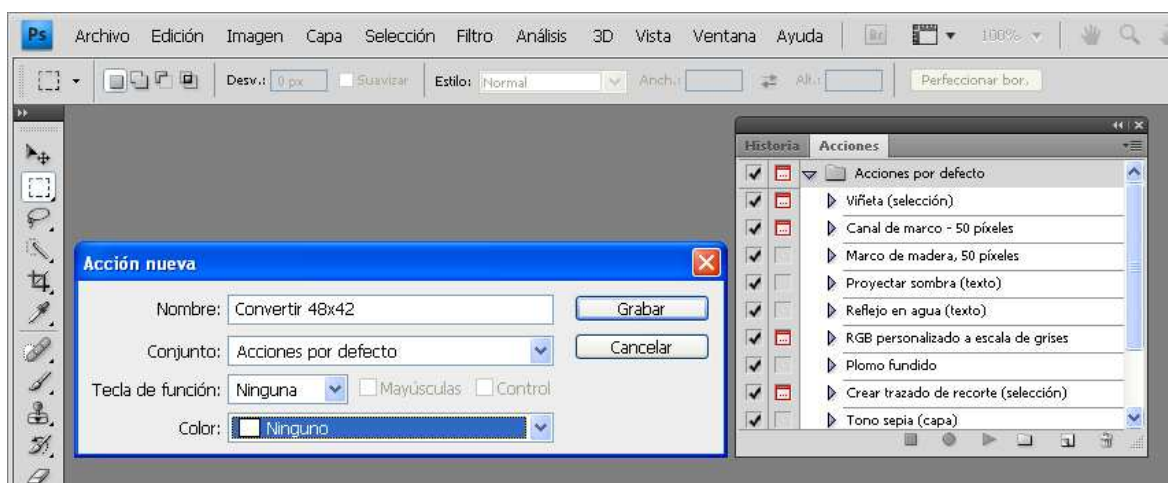


Figura III.1 – Crear nova acció

Es crea una nova acció a través de l'ícona *Crear nova acció* de l'eina *Accions* i se li dóna un nom. A partir d'aquí, s'estan grabant automàticament els passos que es duen a terme dins l'aplicació fins que no es decideixi guardar el procés.

El següent pas és dirigir-se altra vegada a la barra d'eines superior i seleccionar l'eina *Tamany d'imatge* de la opció *Imatge*. En aquesta finestra caldrà introduir les dimensions que necessiten les imatges que es volen processar i desbloquejar la opció *Restringir proporcions*.

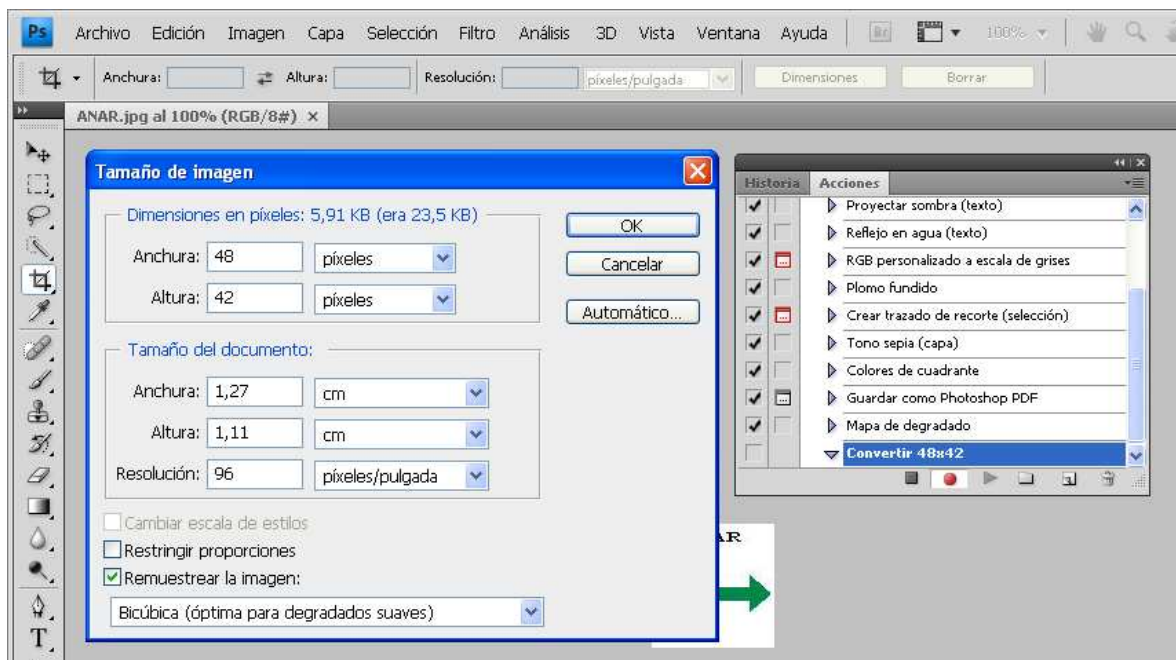


Figura III.2 – Selecció del tamany d'imatge

La imatge passa, doncs, a tenir les dimensions que necessita l'aplicació i l'acció guarda el moviment fet del canvi del tamany de la imatge. Només caldrà un últim pas de guardar la imatge i així l'acció que està grabant ho pugui reproduir en el moment que es sol·liciti l'automatització. En aquest punt, es para la grabació de moviments i ja es disposa de l'acció necessària per redimensionar una imatge amb un tamany seleccionat.

El següent pas, i últim, automatitza l'acció sobre un lot d'imatges d'una carpeta. Cal dirigir-se a l'eina Arxiu/Automatitzar/Lot... i, a la finestra que apareix, seleccionar l'acció creada i la carpeta que conté les imatges a redimensionar.

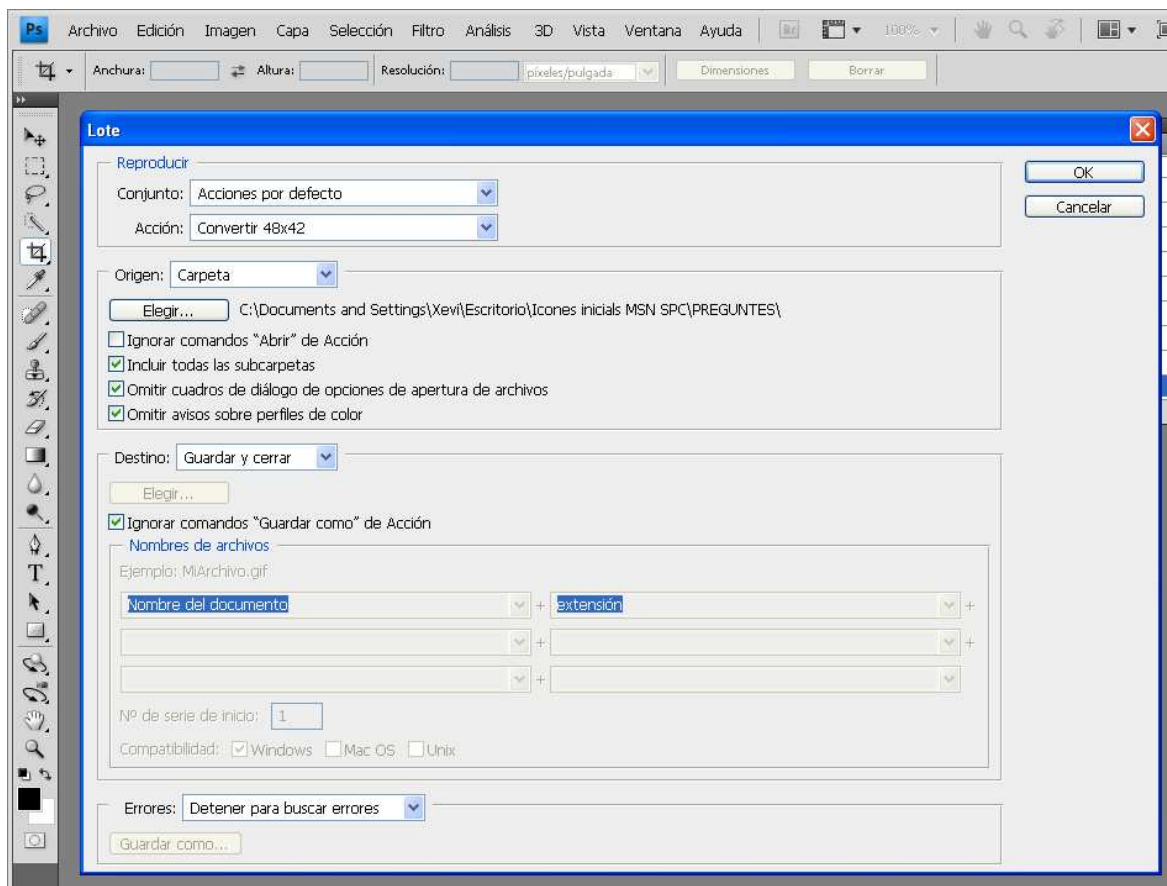


Figura III.3 – Automatització de l'acció

Un cop executada l'automatització, les imatges de la carpeta passaran a tenir les dimensions que se li han assignat en el moment de la grabació de l'acció.

El tamany de les imatges dels símbols es troben en 48x42 píxels i les imatges dels usuaris a 60x60 píxels i 128x128 píxels. Amb aquestes mides, l'aplicació adequa perfectament la imatge al format que té predefinit.

Annex IV – Codi de creació de la base de dades

La sentència de creació de la base de dades és:

```
CREATE DATABASE messenger_visual;
```

Per tal de poder treballar amb aquesta base de dades, cal inserir la sentència:

```
USE messenger_visual;
```

A partir d'aquí, ja es pot inserir el codi de creació de la base de dades següent:

```
/*=====*/
/* DBMS name:      MySQL 5.0                      */
/* Created on:     01/05/2009 22:55:18           */
/*=====*/

drop table if exists PERFILSIMBOL;

drop table if exists SIMBOL;

drop table if exists CATEGORIA;

drop table if exists USUARI;

drop table if exists PERFIL;

/*=====*/
/* Table: CATEGORIA                                */
/*=====*/
create table CATEGORIA
(
  IDCATEGORIA      int not null,
  NOM               varchar(30) not null,
  IMATGE           blob,
  DESCRIPCIO       text,
  primary key (IDCATEGORIA)
);
```

Annex IV – Codi de creació de la base de dades

```

/*=====*/
/* Table: PERFIL */
/*=====*/
create table PERFIL
(
    IDPERFIL          int not null,
    NOM               varchar(30) not null,
    DESCRIPCIO       text,
    primary key (IDPERFIL)
);

/*=====*/
/* Table: PERFILSIMBOL */
/*=====*/
create table PERFILSIMBOL
(
    IDPERFIL          int not null,
    IDSIMBOL          int not null,
    primary key (IDPERFIL, IDSIMBOL)
);

/*=====*/
/* Table: SIMBOL */
/*=====*/
create table SIMBOL
(
    IDSIMBOL          int not null auto_increment,
    IDCATEGORIA       int,
    NOM               varchar(30) not null,
    IMATGE            blob not null,
    DESCRIPCIO       text,
    primary key (IDSIMBOL)
);

/*=====*/
/* Table: USUARI */
/*=====*/
create table USUARI
(
    IDUSUARI          int not null,
    IDPERFIL          int,
    NOM               varchar(30) not null,
    COGNOM1           varchar(20) not null,
    COGNOM2           varchar(20),
    IMATGE            blob not null,
    EMAIL             varchar(50),
    PASSWORD          varchar(20),
    primary key (IDUSUARI)
);

```

```
alter table PERFILSIMBOL add constraint FK_PERFILSIMBOL foreign
key (IDPERFIL)
    references PERFIL (IDPERFIL) on delete restrict on update
restrict;
```

```
alter table PERFILSIMBOL add constraint FK_PERFILSIMBOL2 foreign
key (IDSIMBOL)
    references SIMBOL (IDSIMBOL) on delete restrict on update
restrict;
```

```
alter table SIMBOL add constraint FK_CONTE foreign key
(IDCATEGORIA)
    references CATEGORIA (IDCATEGORIA) on delete restrict on
update restrict;
```

```
alter table USUARI add constraint FK_TE foreign key (IDPERFIL)
    references PERFIL (IDPERFIL) on delete restrict on update
restrict;
```

Una sentència ràpida per veure que les taules s'han creat és:

```
SHOW TABLES;
```


9. Bibliografia

- [1] <http://www.fundmaresme.com/>, Fundació el Maresme Pro Persones Amb Disminució Psíquica
- [2] <http://books.google.es>, Addison-Wesley/Diaz de Santos, *Diseño Conceptual de Bases de Datos*
- [3] Craig Larman, *UML y patrones*. Pearson, Prentice-Hall
- [4] Eliote Rusty Harold, *Java Network Programming*. Ed. O'Reilly 2000.
- [5] Xavier Costa, *Messenger Visual*, 2009
- [6] <http://www.xtec.cat/formaciotic/dvdformacio/materials/td116/d116m4/d116m4p1.htm>, *Creació i gestió d'entorns web dinàmics*
- [7] <http://blocs.xtec.cat/educacioespecialssdh/2008/04/10/el-sistema-spc-de-comunicacio>, *El sistema de comunicació SPC*
- [8] <http://es.wikipedia.org/wiki/Wikipedia:Portada>, Wikipedia, la enciclopedia libre

