
PROJECTE DE FINAL DE CARRERA



UNIVERSITAT
ROVIRA I VIRGILI



ESCOLA
TÈCNICA
SUPERIOR
D'ENGINYERIA

Implementació d'un Sistema Multi Agent per a la gestió d'una Unitat de Cures Pal·liatives

Marta Marín Torrell

marta.marin@estudiants.urv.cat

Enginyeria en Informàtica

Dirigit per el **Dr. Antonio Moreno**

Escola Tècnica Superior d'Enginyeria (ETSE)

Universitat Rovira i Virgili (URV)

<http://www.etse.urv.cat>

Curs 2005-2006

ÍNDIX

1	INTRODUCCIÓ.....	7
1.1	PALLIASYS	7
1.1.1	Els objectius principals del projecte Palliasys.....	8
1.1.2	Arquitectura de PalliaSys.....	8
1.2	DESCRIPCIÓ DEL PROJECTE.....	9
1.3	ORGANITZACIÓ DEL DOCUMENT	10
2	SISTEMES MULTI AGENT (SMA).....	13
2.1	DEFINICIÓ D'UN SMA	13
2.2	AVANTATGES D'UN SMA.....	13
2.3	ESPECIFICACIÓ FIPA DE LA GESTIÓ D'AGENTS.....	14
2.4	EL LLENGUATGE DE COMUNICACIÓ ENTRE AGENTS	15
2.5	PERQUÈ UN SISTEMA MULTI AGENT?.....	17
2.5.1	Modularitat.....	18
2.5.2	Eficiència.....	18
2.5.3	Fiabilitat.....	18
2.5.4	Flexibilitat	19
3	ARQUITECTURA DEL SISTEMA.....	21
3.1	APLICATIU WEB	21
3.1.1	Sistema telemàtic per a l'assistència mèdica d'una UCP	21
3.1.2	Modificacions	22
3.2	SISTEMA MULTI AGENT	23
3.2.1	Agent Pacient.....	24
3.2.1.1	Cicle de vida	24
3.2.1.2	Behaviours	25
3.2.2	Agent Socket	26
3.2.2.1	Behaviours	27
3.2.3	Agent DBWrapper	27
3.2.3.1	Behaviours	28
3.2.4	Agent Metge.....	28
3.2.4.1	Behaviours	29
3.2.5	Agent Analitzador	29
3.2.5.1	Behaviours	30
3.2.6	Comunicació entre Agents.....	30
3.2.6.1	Subscripció d'Alarmes (FIPA-Subscribe).....	30
3.2.6.2	Petició a la Base de Dades (FIPA Request Protocol).....	31
3.3	BASE DE DADES.....	32
3.3.1	Introducció	32
3.3.2	Model Relacional.....	33
3.3.3	Taules	33
3.3.3.1	Taules dels usuaris del sistema.....	34
3.3.3.2	Taules relacionades amb les Alarmes.....	34
3.3.3.3	Taules relacionades amb les dades mèdiques dels pacients	35
3.4	SEGURETAT	36
3.4.1	Requeriments de seguretat	36
3.4.1.1	Requeriments que afecten directament a l'aplicatiu.....	36

3.4.1.2	Requeriments de responsabilitat per part del personal de la UCP.....	37
3.4.2	<i>Mesures implementades</i>	37
3.4.2.1	Perfils d'usuari i Autenticació.....	38
3.4.2.2	Registres d'Accés: Fitxers de log.....	38
3.4.2.3	Comunicació Segura.....	39
4	GESTIÓ AUTOMATITZADA D'ALARMES	41
4.1	AUTO AVALUACIÓ.....	41
4.2	TIPUS D'ALARMES.....	42
4.2.1	<i>Alarmes Bàsiques</i>	43
4.2.2	<i>Alarmes d'Evolució</i>	44
4.3	ALARMES: GESTIÓ I CICLE DE VIDA.....	46
4.3.1	<i>Definició</i>	46
4.3.2	<i>Assignació</i>	48
4.3.3	<i>Validació</i>	49
4.3.4	<i>Activació</i>	49
5	INSTAL·LACIÓ / MANUAL DE L'ADMINISTRADOR	51
5.1	REQUERIMENTS DEL SISTEMA.....	51
5.2	INSTAL·LACIÓ DEL PROJECTE DEL PABLO.....	51
5.2.1	<i>El Servidor Web Apache</i>	51
5.2.2	<i>Mòdul SSL sobre Apache</i>	52
5.2.2.1	Primera configuració sobre Apache.....	52
5.2.2.2	Instal·lació del mod_ssl i OpenSSL.....	52
5.2.2.3	Creació d'un certificat de test.....	53
5.2.2.4	Configuració d'Apache i mod_ssl.....	53
5.2.3	<i>PHP sobre Apache</i>	54
5.2.3.1	Modificacions sobre PHP.....	54
5.2.4	<i>Base de Dades i Connexió ODBC</i>	54
5.2.5	<i>Instal·lació de l'Aplicació Web</i>	55
5.3	INSTAL·LACIÓ DEL SISTEMA MULTI AGENT.....	55
5.4	POSTA EN MARXA / EXECUCIÓ.....	56
5.4.1	<i>Servidor (Agent DBWrapper, Agent Socket i Agents Pacient)</i>	56
5.4.2	<i>Agent Metge</i>	56
5.4.3	<i>Agent Estadístic</i>	56
6	INTERFÍCIE GRÀFICA / MANUAL D'USUARI	57
6.1	AGENT METGE.....	57
6.1.1	<i>Gestió dels Agents Pacient</i>	59
6.1.1.1	Alta Agent Pacient.....	59
6.1.1.2	Baixa Agent Pacient.....	60
6.1.2	<i>Generació d'Alarmes</i>	60
6.1.2.1	Generació d'una Alarma Bàsica.....	61
6.1.2.2	Generació d'una Alarma d'Evolució.....	64
6.1.3	<i>Assignació d'Alarmes</i>	65
6.1.4	<i>Alarmes Actives</i>	66
6.1.5	<i>Historial d'Alarmes</i>	67
6.2	AGENT ESTADÍSTIC.....	68
6.2.1	<i>Opcions generals</i>	70
6.2.1.1	Definir el Període.....	70
6.2.1.2	Guardar Estadístiques i Gràfiques.....	70

6.2.2	<i>Estadístiques</i>	70
6.2.3	<i>Gràfiques</i>	74
6.2.4	<i>Alarmes</i>	76
7	JOC DE PROVES	77
7.1	VALIDACIONS D'ACCÉS.....	77
7.2	ALTA DELS AGENTS PACIENT	77
7.3	ALARMES	78
7.4	DADES ESTADÍSTIQUES	79
7.4.1	<i>Estadístiques en HTML</i>	79
7.4.2	<i>Gràfiques</i>	79
7.5	SEGURETAT - FITXETRS DE LOG	79
8	CONCLUSIONS I TREBALL FUTUR	81
8.1	CONCLUSIONS	81
8.2	TREBALL FUTUR.....	81
ANNEX A	IMPLEMENTACIÓ D'UN SMA EN JADE	83
A.1	PLATAFORMA D'AGENTS DE JADE.....	83
A.2	IMPLEMENTACIÓ D'AGENTS.....	83
A.3	ELS PAQUETS DE JADE	84
A.4	ENTORN GRÀFIC DEL JADE.....	84
	REFERÈNCIES I BIBLIOGRAFIA	86

ÍNDIX DE FIGURES

FIGURA 1 - ARQUITECTURA PALLIASYS	8
FIGURA 2 – MODEL D'ARQUITECTURA D'UN SMA PER LA FIPA.....	14
FIGURA 3 - ESTRUCTURA BÀSICA D'UN MISSATGE ACL.....	15
FIGURA 4 - ARQUITECTURA DEL SMA	23
FIGURA 5 - ESQUEMA DEL PROTOCOL DE SUBSCRIPCIÓ	31
FIGURA 6 - ESQUEMA DEL REQUEST PROTOCOL	32
FIGURA 7 - MODEL RELACIONAL DE LA BASE DE DADES	33
FIGURA 8 - SEGURATAT, PANTALLA DE VALIDACIÓ	38
FIGURA 9 - PANTALLA D'ENTRADA DE LES AUTO AVALUACIONS	42
FIGURA 10 - EXEMPLE D'ESTRUCTURACIÓ D'UNA ALARMA	44
FIGURA 11 - PANTALLA EDICIÓ ALARMES BÀSIQUES	46
FIGURA 12 – PANTALLA EDICIÓ ALARMES EVOLUCIÓ.....	47
FIGURA 13 - PANTALLA ASSIGNACIÓ ALARMES	48
FIGURA 14 - FINESTRA DE VALIDACIÓ	57
FIGURA 15 - ERROR DE VALIDACIÓ	57
FIGURA 16 - AGENT METGE	58
FIGURA 17 - PANTALLA ALTA AGENT PACIENT	59
FIGURA 18 - MISSATGE ERROR EN LA CREACIÓ DE L'AGENT PACIENT	60
FIGURA 19 - PANTALLA BAIXA AGENT PACIENT.....	60
FIGURA 20 - PANTALLA DE GENERACIÓ D'ALARMES CONDICIONALS	61
FIGURA 21 - DETALL PANTALLA ALARMES BÀSIQUES: NOM ALARMA.....	62
FIGURA 22 - DETALL PANTALLA ALARMES BÀSIQUES: DEFINICIÓ CONDICIONS	62
FIGURA 23 - DETALL PANTALLA ALARMES BÀSIQUES: ZONA PISSARRA	62
FIGURA 24 - ERROR A L'HORA DE COMBINAR CONDICIONS	63
FIGURA 25 - PANTALLA IMPORTAR ALARMES BÀSIQUES	63
FIGURA 26 – DETALL PANTALLA ALARMES BÀSIQUES: CONDICIÓ SELECCIONADA	64
FIGURA 27 - PANTALLA DE GENERACIÓ D'ALARMES D'EVOLUCIÓ.....	64
FIGURA 28- PANTALLA D'ASSIGNACIÓ D'ALARMES	65
FIGURA 29 - MENÚ AFEGIR ALARMA	66
FIGURA 30 - AVÍS ALARMA ACTIVADA	66
FIGURA 31 - COMPTADOR ALARMES ACTIVES	66
FIGURA 32 - PANTALLA ALARMES ACTIVES.....	67
FIGURA 33 - HISTORIAL ALARMES	68
FIGURA 34 - VALIDACIÓ ADMINISTRADOR.....	68
FIGURA 35 - AGENT ESTADÍSTIC	69
FIGURA 36 - PANTALLA DE DEFINICIÓ DEL PERÍODE	70
FIGURA 37 - DADES A SELECCIONAR PER A MOSTRAR A L'INFORME	71
FIGURA 38 - EXEMPLE DE GRÀFICA DE BARRES	75
FIGURA 39 - EXEMPLE DE GRÀFICA DE PASTÍS.....	75
FIGURA 40 - JADE REMOTE AGENT MANAGEMENT GUI	85

1 Introducció

En un hospital, la **Unitat de Cures Pal·liatives (UCP)** és qui té la funció d'ocupar-se dels malalts terminals, amb la finalitat de pal·liar, en el major grau possible, el dolor i el patiment que puguin tenir en l'etapa final de la seva vida. Totes les persones involucrades en una UCP, tant els pacients com els facultatius que hi treballen, tenen una sèrie de necessitats de manipulació d'informació en un entorn distribuït. A banda, els metges també requereixen mantenir un cert control dels pacients i la seva evolució, tant si aquests es troben ingressats a la Unitat com si són externs, en aquest últim cas, que és el més usual, aquest fet dificulta que aquest seguiment sigui acurat.

En l'any 1977, en l'*Hospital Royal Victoria* de Canadà va aparèixer el concepte de **medicina pal·liativa** que es podia definir com una “assistència sanitària total, activa i continuada dels pacients en fase terminal i de la seva família per part d'un equip interdisciplinari amb l'objectiu de millorar la qualitat de vida sense la intenció d'allargar la supervivència”. Uns deu anys més tard, a Espanya es proposa un model de medicina pal·liativa basada en l'atenció dels pacients dintre de l'hospital, a càrrec del personal de l'hospital, i a domicili a càrrec dels **PADES** (*Programa d'Atenció Domiciliària – Equips de Suport*) i afegint-se en últim lloc la visita del pacient a els metges de la unitat de cures pal·liatives d'un hospital.

El projecte que presentem a continuació forma part del projecte **Palliasys** [1] que es troba format per dos grans blocs, el primer és un sistema basat en *Tecnologies de la informació i les Comunicacions (TIC)* que està pensat per tal de poder satisfer les necessitats de recollida i consulta d'informació dels pacients i dels metges de la **Unitat de Cures Pal·liatives de l'Hospital de la Santa Creu i Sant Pau de Barcelona**. El segon està format bàsicament per el projecte que ens ocupa i consta d'un **Sistema Multi Agent** que ajuda a monitoritzar l'evolució dels pacients mitjançant un sistema d'Alarmes que s'explicarà detalladament en apartats següents.

1.1 Palliasys

L'objectiu principal del projecte **Palliasys** [1] es dissenyar, construir i desplegar un sistema automatitzat per a millorar la gestió i tractament de les dades emmagatzemades en la unitat de Cures Pal·liatives (UCP) d'un Hospital gran. Aquesta unitat, com s'acaba de comentar, està especialitzada en tractar gent amb malalties terminals i intentar ajudar-los a tenir una qualitat mínima de vida. Degut a la tipologia d'aquesta atenció mèdica, la majoria de pacients es troben a casa, cosa que dificulta el seu seguiment

El sistema **Palliasys** està especialment dirigit als pacients que romanen a casa seva, d'aquesta manera tenen la possibilitat de tenir accés a alguns serveis sense tenir que desplaçar-se a l'hospital, fet que, degut al seu estat, normalment sol ser desitjable evitar-ho al màxim. També pretén ser una eina útil per al personal mèdic de la UCP que d'aquesta forma pot realitzar un seguiment d'aquests pacients d'una forma molt més acurada gràcies a que el sistema incorpora utilitats com les d'Anàlisi de dades o d'Alarmes.

1.1.1 Els objectius principals del projecte *Palliasys*

Un dels primers objectius del sistema *PalliaSys* és el de millorar el procés de recopilar i recollir informació dels pacients pal·liatius. Aquestes dades s'emmagatzemen en una Base de dades que es troba en la UCP del centre mèdic. El sistema proporciona un accés segur i autènticat a les dades dels pacients.

La segona meta del projecte és la de millorar la informació de la que disposen els doctors sobre els pacients. Amb la utilització d'un **Sistema Multi Agent (SMA)** és possible supervisar contínuament l'estat de tots i cada un dels pacients i proactivament proporcionar informació personalitzada, detallada i actualitzada als doctors de la UCP. Aquest és l'objectiu principal d'aquest Projecte Final de Carrera.

Finalment, una altra meta important de *PalliaSys* és la de realitzar un anàlisi intel·ligent de les dades històriques recopilades en la UCP. Aquesta tasca també es trobarà integrada en el Sistema Multi Agent i serà portada a terme per l'*Agent Analitzador*.

1.1.2 Arquitectura de *PalliaSys*

L'arquitectura bàsica del sistema *PalliaSys* es mostra (en una forma simplificada) en la **Figura 1**.

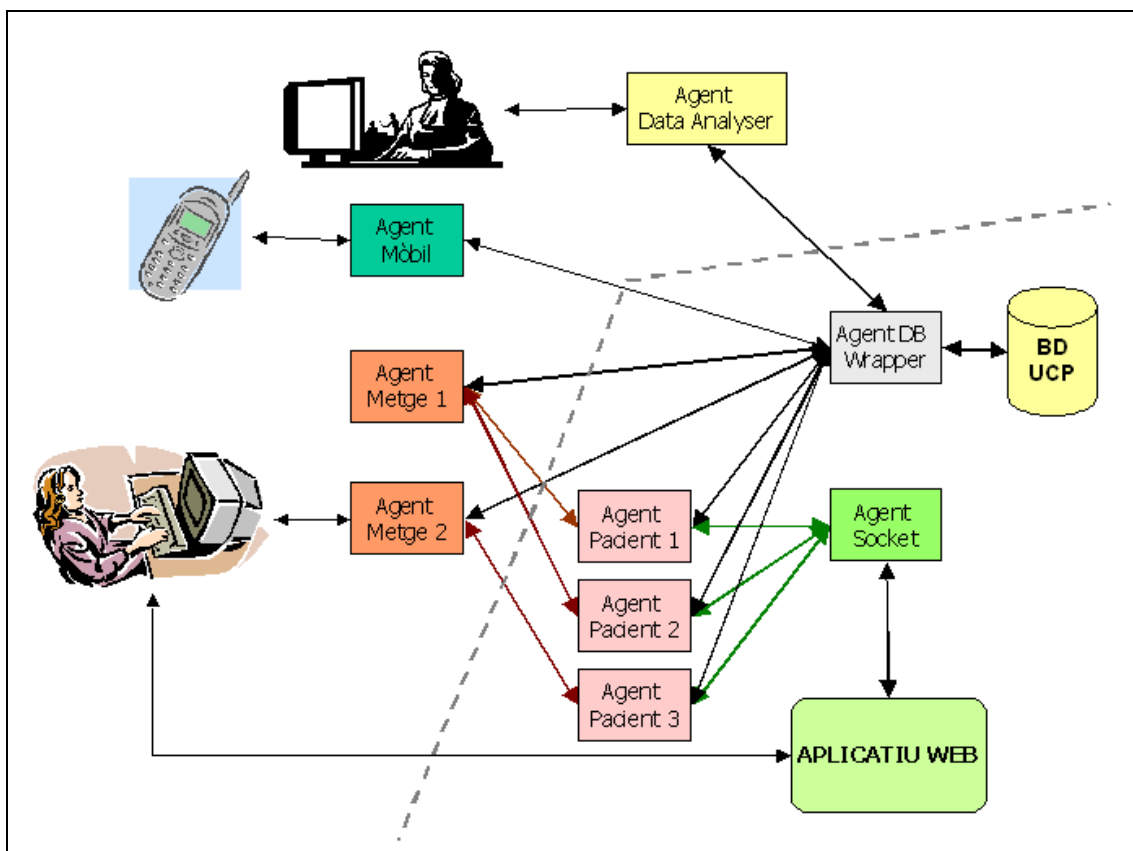


Figura 1 - Arquitectura PalliaSys

Dintre del projecte *PalliaSys* podem distingir dos parts principals:

1. **Tecnologies de la informació i les comunicacions**, fetes servir per els pacients a casa seva per a proporcionar o consultar les seves dades mèdiques personals, per a enviar auto avaluacions o per a rebre informació dels doctors.
2. **Un sistema multi agent**, emprat per a manejar amb seguretat les dades clíniques, sense perdre de vista l'estat dels pacients pal·liatius i analitzant la seva evolució.

En el sistema podem trobar sis tipus d'agents, els que formin part d'aquest Projecte Final de Carrera s'explicaran més detalladament en l'apartat d'arquitectura d'aquest document.

- **Agent Wrapper de la Base de Dades**: Aquest agent controla l'accés a la Base de Dades de la Unitat de Cures Pal·liatives.
- Un **Agent Metge** per a cada metge de la UCP. Aquest agent funciona en l'ordinador personal de cada metge i serà l'encarregat de proporcionar l'interfície amb el metge perquè manegui els *Agents Pacient* i el sistema d'alarmes.
- Un **Agent Pacient** per a cada pacient. Aquests agents seran els responsables de controlar l'evolució del pacient.
- Un **Agent Analitzador de Dades** especialitzat en mineria de dades, el descobriment del coneixement i tècniques d'aprenentatge automàtic. A banda, també ens permetrà la generació d'informes amb estadístiques i gràfiques. Aquest agent també serà l'encarregat de definir *Alarmes Globals* si és engegat amb els permisos del cap de la UCP.
- Un **Agent Socket** que permet la comunicació entre les dues parts de l'Arquitectura del sistema **Palliasys**. Aquest agent serà l'encarregat d'avisar als agents pacient per a que realitzin la tasca de comprovació de les *Alarmes* amb la conseqüent activació si és necessari.
- Un **Agent que s'executarà en entorns mòbils**. Aquest agent ens permet interactuar amb el sistema mitjançant un aparell de telefonia mòbil, podent enviar una auto avaluació des de qualsevol lloc on es trobi el pacient.

1.2 Descripció del Projecte

En aquest projecte, tal i com s'ha comentat, s'ha definit i implementat la major part del Sistema Multi Agent del projecte **Palliasys**. Concretament, en aquest projecte s'ha definit i implementat els següents tipus d'agents comentats en la secció anterior: **Agent Wrapper**, **Agent Metge**, **Agent Pacient**, **Agent Analitzador** (els informes amb estadístiques, les gràfiques i la generació d'Alarmes Globals) i l'**Agent Socket**.

L'objectiu principal d'aquest projecte és el d'oferir un complet sistema d'**Alarmes** que permeti als Metges portar un control acurat de l'evolució dels pacients. És a dir, un sistema que els hi permeti poder controlar els casos en els que, per exemple, un pacient

pateixi de cop i volta un fort dolor o poder detectar un ràpid increment o decrement de les nàusees. Cada metge, mitjançant l'**Agent Metge**, podrà definir quins son els casos que més li interessa de detectar mitjançant la creació d'una sèrie d'Alarmes que podrà assignar als pacients (**Agents Pacient**) que tingui que seguir. Aquest sistema d'Alarmes és de gran utilitat sobretot pels casos en els que els pacients no es troben ingressats en la UCP. Gràcies a ell es pot detectar ràpidament un cas en el que el pacient hagi patit una evolució preocupant en els seus símptomes en els últims dies, o bé que s'hagi donat un canvi brusc que mereixi l'atenció del Metge. En el cas de que els pacients es trobin internats, això no és complicat, però en el cas de que es trobin fora de la UCP és difícil de controlar i d'aquesta forma cobrim aquesta necessitat.

Per al control de les alarmes disposem de dos perfils, el del *Metge* a través de l'**Agent Metge** i el de l'*Administrador* a través de l'**Agent Analitzador**. El Rol del Metge ens permetrà definir i assignar les alarmes als pacients (**Agents Pacient**), així com rebre l'avís de que s'ha activat una alarma. Les alarmes definides per cada metge només seran visibles per a ell mateix. En canvi, administrador defineix alarmes que seran accessibles per tots els metges. D'aquesta forma es podrà disposar d'una sèrie d'Alarmes comuns per a tots els Metges.

Una altra funcionalitat ja esmentada és la de tractament de dades a través de la creació d'estadístiques i gràfiques per l'**Agent Analitzador**. D'aquesta forma es podrà realitzar un estudi sobre quina és la tendència dels casos que es porten a la UCP. Aquestes estadístiques i gràfiques seran d'utilitat a nivell del cap de Unitat de Cures Pal·liatives. A banda, el projecte *Palliasys* recordem que també disposa d'algorismes d'anàlisi de dades que s'han definit i implementat en el projecte del Raul Mateos [2].

L'Accés a la Base de Dades es centralitza tot en l'**Agent Wrapper de la Base de Dades (BD)**. Tots els agents es comunicaran amb aquest agent per obtenir dades de la BD o per a modificar-la.

Per a poder ferir tota aquestes funcionalitats, el projecte s'ha hagut d'integrar amb el projecte del Pablo García Rué [3] que és el que correspon a la part del sistema basat en *Tecnologies de la informació i les Comunicacions*. Els punts de confluència dels dos projectes són la Base de Dades i la comunicació a través de Sockets (**Agent Socket**). Per tal d'aconseguir aquesta integració, també s'ha hagut de modificar parts del projecte del Pablo per tal de possibilitar la correcta comunicació i emmagatzematge de dades. Al mateix temps, també s'ha aprofitat per a realitzar una sèrie de modificacions que s'havien requerit des de la Unitat de Cures Pal·liatives.

1.3 Organització del Document

Després d'aquesta introducció, en el document es podria agrupar la informació en els quatre grans blocs següents:

1. **Definició SMA i JADE:** En el primer trobarem una explicació sobre què és un Sistema Multi Agent, en aquest apartat ens centrarem en l'estàndard que ha definit la *FIPA - Foundation for Intelligent Physical Agents* [4]. En aquest bloc també hi ha un apartat en el que es parla sobre el *JADE - Java Agent DEvelopment framework*. Aquest serà l'entorn de treball (*Framework*) que es

farà servir en aquest projecte per tal de facilitar les tasques de desenvolupament del **Sistema Multi Agent**. Ni que formi part d'aquest bloc, aquest apartat el trobarem com un annex.

2. **Disseny:** En aquest bloc es parlarà del projecte del Pablo García Rué [3] i les modificacions realitzades en aquest projecte. S'explicarà detalladament com està estructurat el projecte que ens ocupa, els diferents *Agents* per als que està compost el *Sistema Multi Agent*. Es presentarà un disseny simplificat de la Base de Dades i es detallaran les mesures de seguretat que s'han aplicat. També es prestarà especial atenció a les Alarmes, per això s'ha dedicat un apartat específic per a elles.
3. **Aplicació:** En l'última part del document ens centrarem en el treball realitzat en aquest Projecte. Primer, el *Manual de l'Administrador*, es detallaran els passos a seguir per a instal·lar l'aplicació i una mica d'explicació de com configurar l'aplicació. Seguidament tindrem l'apartat en el que es veurà el funcionament del projecte amb el *Manual d'Usuari*. Aquest bloc s'acaba de complementar amb el *Joc de Proves*, en el que veurem exemples i proves dels diferents utilitats de l'aplicació.
4. **Conclusions i Bibliografia:** Per acabar tancarem la documentació amb les conclusions en les que es farà una reflexió sobre el projecte i el possible *treball futur*. Al final es trobarà la Bibliografia amb els documents, projectes i entitats dels que s'ha extret informació o fet referència al llarg d'aquesta documentació.

2 Sistemes Multi Agent (SMA)

Per tal de facilitar entendre l'estructura i funcionament d'aquest projecte, en aquest apartat es farà una introducció als **Sistemes Multi Agent**, com són, quines avantatges tenen i també s'explicarà perquè s'ha optat per a fer servir un SMA.

2.1 Definició d'un SMA

Un **Sistema Multi Agent** està format per una sèrie d'agents que cooperen, es comuniquen i es coordinen entre ells amb la finalitat d'assolir un objectiu comú.

Hi ha moltes definicions diferents d'agent, però el podríem definir com un sistema que habita en un entorn complex i dinàmic; és capaç de sentir i actuar de manera autònoma en el seu entorn, i té un conjunt d'objectius o motivacions que intentarà aconseguir a través de les seves accions [Wooldridge, 2002].

En aquest projecte seguirem el model de gestió d'agents definit per la **FIPA** (*Foundation for Intelligent Physical Agents*) [4].

2.2 Avantatges d'un SMA

La utilització d'un Sistema Multi Agent ens proporciona una gran quantitat d'avantatges enfront la programació tradicional. A continuació passem a enumerar els més importants i a fer-me una breu explicació.

- **Modularitat:** Aquesta és potser la propietat més important i útil de la que podem disposar i la que genera la resta d'avantatges. Quan parlem de Modularitat ho fem des del punt de vista de la divisió de tasques i rols entre els diferents agents. Aquesta separació ens permet reduir la complexitat de les tasques obtenint una programació més estructurada. A banda, també ens permet la reutilització ràpida i eficient de les funcionalitats ja que cada Agent és com un mòdul independent de codi que ens permetrà redistribuir els agents en les diferents plataformes aconseguint noves configuracions.
- **Eficiència:** Aquest avantatge deriva directament de l'anterior i ve de que el fet de poder repartir les tasques entre varis agents permet paral·lelisme quan aquests estan treballant en màquines diferents.
- **Fiabilitat:** En un SMA si un dels seus elements deixa de funcionar, no vol dir que aquest hagi de deixar de fer-ho. A més replicant serveis i provocant així una redundància es pot aconseguir més seguretat
- **Flexibilitat:** A més de tot el que s'ha dit, en un Sistema Multi Agent es poden incorporar i eliminar pacients de forma dinàmica sense causar per això que aquest deixi de funcionar.

2.3 Especificació FIPA de la gestió d'agents

La **FIPA** (*Foundation for Intelligent Physical Agents*) [4] ens proporciona un estàndard per on es defineix el cicle de vida dels agents [8], és a dir el model a seguir per a la creació, registre, comunicació, establiment, mobilitat i destrucció d'agents. El model proposat el podem contemplar en la **Figura 2**.

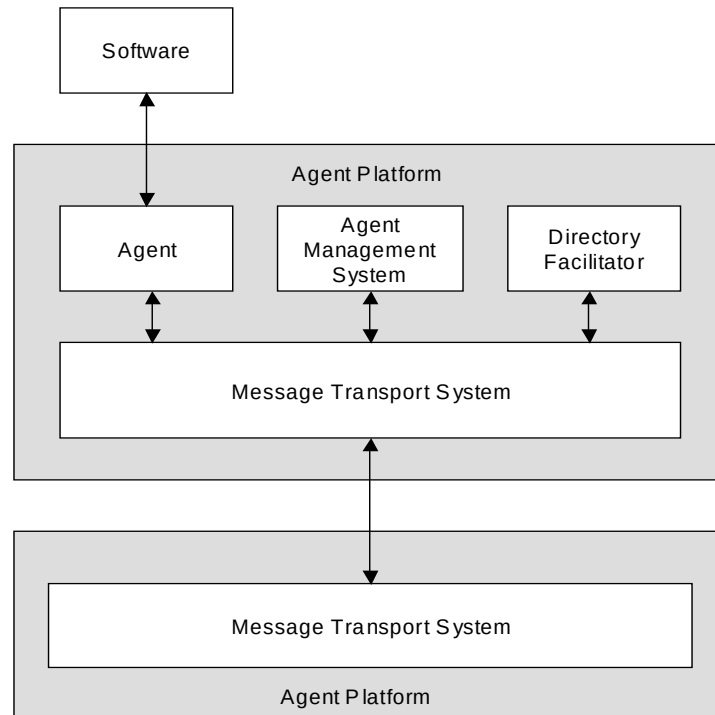


Figura 2 – Model d'arquitectura d'un SMA per la FIPA

En aquest esquema podem observar els elements dels que, segons la FIPA, ha d'estar compost un Sistema Multi Agent[8]:

- **Agents:** Considerats com la Unitat Bàsica del sistema, es podria definir com un programa que encapsula una sèrie de serveis.
- **Plataforma d'Agents (Agent Platform):** És la base que ens sobre la que es crearan i executaran els diferents agents, ens proporciona els mitjans necessaris per a que això es pugui dur a terme.
- **Agent DF (Directory Facilitator):** Aquest agent ens ofereix la localització dels serveis que ens ofereixen els demés agents dels SMA. Podria dir que realitza un servei de *Pàgines Grogues*. Per tal de que els seus serveis puguin ser oferts per l'Agent DF, la resta d'agents s'hi han de registrar.
- **Agent AMS (Agent Management System):** Es tracta d'un agent únic en una plataforma d'agents (AP). La seva funció és al de portar a terme un control de supervisió de l'accés i ús de la plataforma. Té emmagatzemats els noms lògics i les adreces de transport dels agent i ofereix un servei de

Pàgines Blanques.

- **Message Transport Service (MTS):** És el mecanisme emprat pels agents per a comunicar-se entre ells tant si es troben en la mateixa plataforma, com si ho han de fer remotament.
- **Software:** Pot estar compost per les utilitats o programes que estan relacionats amb el SMA i interactuen amb ell oferint una sèrie de funcionalitats, però que no formen part d'ell.

2.4 El llenguatge de comunicació entre agents

Per separat, els agents poden oferir una certa utilitat, però el seu veritable potencial apareix quan cooperen entre ells per tal d'arribar a assolir un determinat objectiu comú. Aquesta és la filosofia d'un *Sistema Multi Agent*. El coneixement d'un agent sol ser limitat, a vegades fins i tot erroni o incomplet, és per això que és necessari que cooperi amb altres agents per tal de poder obtenir un benefici i aconseguir els seus objectius. Per tal de comunicar-se, els agents empren un llenguatge comú l'ACL (*Agent Communication Language*)[6]

El *Llenguatge ACL*[6] està basat en la teoria de la parla. És a dir s'enfoca des del punt de vista que les comunicacions individuals entre els agents es poden reduir a un petit nombre de conceptes o actes comunicatius. Aquests actes són els que donen forma al significat bàsic de la comunicació.

La unitat bàsica d'aquesta comunicació és el missatge. Com en la gestió d'agents, la FIPA defineix l'estructura que tindrà el missatge així com la seva utilització [4]. En la **Figura 3** podem veure l'estructura bàsica d'un missatge.

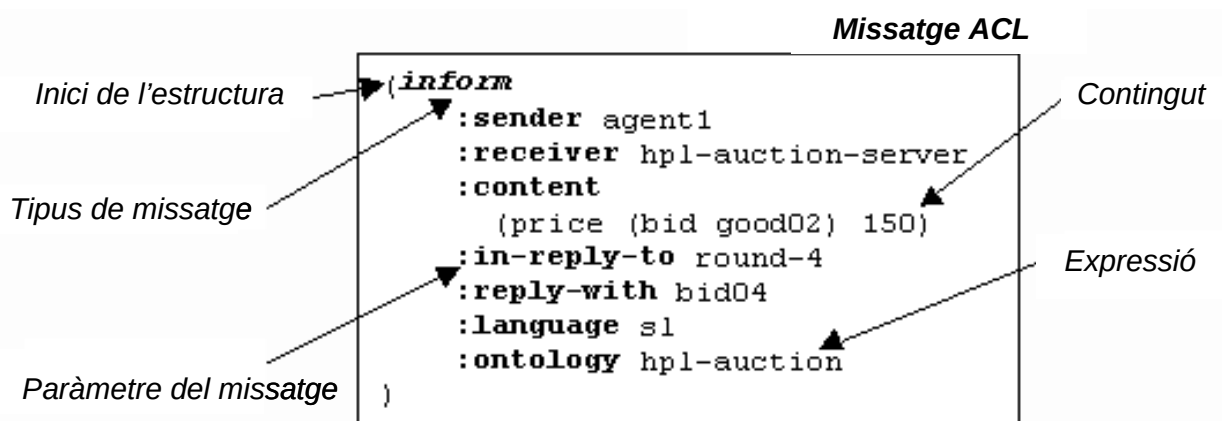


Figura 3 - Estructura bàsica d'un Missatge ACL

Per a entendre millor quin significat té cada part, a continuació es farà una breu explicació de la funció que realitzen:

- **Tipus de missatge:** Dins del tipus de missatge es defineix la *performative* (acció) que ens indicarà quin és l'acte de comunicació que porta a terme aquest missatge. Dintre de les *performatives* més emprades podríem ressaltar-ne unes quantes:
 - **agree:** Acció que indica l'acceptació per part de l'emissor de realitzar una acció que s'ha demanat des de l'emissor.
 - **cancel:** Ens indica que s'ha de portar a terme la cancel·lació d'alguna petició anterior que encara no ha finalitzat.
 - **failure:** Amb aquesta acció s'informa de la fallada d'una petició feta amb anterioritat que no s'ha pogut fer.
 - **inform:** Amb aquesta acció s'informa al receptor que una acció s'ha realitzat correctament.
 - **inform-if i inform-ref:** S'empra per tal d'informa al receptor de si una proposició és certa o no en el cas del *inform-if*, i de si un objecte al que es fa referència és cert o no en el cas del *inform-ref*.
 - **not-understood:** Amb el *not-understood* s'està informat a l'agent receptor que l'agent que emet el missatge no s'ha entès una petició realitzada anteriorment.
 - **query-if i query-ref:** Amb aquestes accions el que es fa és preguntar a un altre agent o bé si un fet és cert en el cas del *query-if*, o sobre un objecte en el cas del *query-ref*.
 - **refuse:** S'està dient que es refusa una petició de realitzar una acció que se li ha demanat de portar a terme. Amb aquesta acció de rebuig també s'especificarà el motiu pel que s'ha refusat.
 - **request:** Amb aquesta acció l'emissor està demanant al receptor que realitzi una determinada acció.
 - **subscribe:** És com una subscripció en la que l'emissor demana que cada cop que es compleixi una determinada condició se l'avisí.
- **Participants (sender / receiver / reply-to):** En aquests camps és on s'indica l'Agent Emissor (*sender*) i l'Agent Receptor (*receiver*). El camp de *reply-to* és opcional i ens indica qui ha de rebre la resposta del missatge en el cas de que es tracti d'un agent diferent a l'emissor.
- **Contingut:** En el contingut del missatge és on es troba el missatge pròpiament dit, és a dir, el que es vol comunicar, l'acció a realitzar en el cas del *request*, la pregunta a fer en el cas del *query*. Aquest contingut pot estar expressat de la forma que es desitgi, és a dir pot ser en text pla, en *html* o en el que es desitgi. Tot i això, i per a seguir amb la línia, aquí s'explicarà quins tipus de contingut

ha definit la FIPA, que de fet, és que s'ha fet servir en el projecte. Així doncs ens trobem que la FIPA ha dividit els continguts dels missatges en quatre possibles categories, creant així quatre tipus de continguts predefinits que es faran servir en funció de les necessitats:

- **FIPA-CCL (FIPA *Constraint Choice Language*):** En aquest tipus es disposa de la definició d'una semàntica que permet l'especificació de predicats amb restriccions.
- **FIPA-SL (FIPA *Semantic Language*):** Potser és el més versàtil i pràctic de tots quatre tipus degut a que és el més general dels quatre i això fa que es pugui aplicar en molts dominis. Permet la formació d'expressions lògiques, l'intercanvi de coneixement de forma òptima i l'expressió d'accions a realitzar.
- **FIPA-KIF (FIPA *Knowledge Interchange Format*):** Aquest és bastant limitat en comparació al FIPA-SL i més especialitzat ja que només es permet expressar objectes com a termes i proposicions com a sentències i no deixa expressar accions.
- **FIPA-RDF (FIPA *Resource Description Framework*):** Aquest és molt similar al FIPA-SL ja que també ens permet la representació d'objectes i les interaccions i accions que poden dur-se a terme entre ells.

En aquest projecte farem servir el FIPA-SL [7] degut a que no es requereix una especialització excessiva i en canvi serà útil el que ofereix aquesta elecció.

- **Paràmetres del missatge:** Aquí és on s'especifica el llenguatge que es farà servir i la ontologia.
- **Expressió:** També és un dels paràmetres que tenen en un missatge. Defineix el vocabulari usat en el contingut dels missatges. És a dir, ens indica l'ontologia que farem servir.

2.5 Perquè un Sistema Multi Agent?

Tenint en compte tots els avantatges proporcionats per un *Sistema Multi Agent*, és fàcil d'entendre per què s'ha escollit per a aquest projecte. Un dels principals requeriments era que ens trobem en un entorn distribuït en que tindrem l'aplicació dividit en diverses màquines. Així doncs, la modularitat proporcionada per un SMA ens proporciona una bona base i moltes facilitats a l'hora de definir-ho i implementar-ho. A més, també ens facilitarà la comunicació entre aquestes diferents parts.

En aquest projecte, a més, també serà necessari realitzar la càrrega dinàmica de trossos de codi (*Agents*) de forma remota, això fa que la flexibilitat que podem obtenir d'un SMA ens vagi molt bé i ens simplifiqui enormement aquesta tasca.

Per a veure més clarament aquests punts, a continuació s'explica una mica més extensament, punt per punt, com afecten aquests avantatges a l'aplicació:

2.5.1 Modularitat

Com ja s'ha comentat, aquesta aplicació es troba en un **entorn distribuït**, és a dir, es troba repartida entre diverses màquines i amb diferents funcions depenent de la seva localització. I tot això amb el requeriment i la necessitat d'una bona comunicació i un bon encaix entre les diferents parts. La **modularitat** que ens proporciona un *Sistema Multi Agent* va acompanyada de funcionalitats de comunicació que simplifiquen i ajuden a facilitar la comunicació entre els mòduls permetent que la crida remota de serveis sigui senzilla.

Gràcies a aquesta característica s'ha pogut repartir l'aplicació entre tres blocs principals:

- el de l'**Agent Metge**
- el de l'**Agent Analitzador de Dades**
- i el de **Serveis**.

Els dos primers els podem trobar en més d'una màquina i el tercer el trobarem al servidor i és on tindrem els Agents *invisibles* als usuaris com el **Agent Wrapper de Base de Dades** o *semi-invisibles* com els **Agents Pacient** dels que l'usuari solament controlarà la creació i destrucció. A més, la modularitat d'un SMA ens permet combinar els mòduls com vulguem, podent, si fos necessari, agrupar els tres en una mateixa màquina.

2.5.2 Eficiència

El repartiment de tasques entre diferents agents ens porta cap a una especialització que ens proporciona una major eficiència. És a dir, la **modularització** té com un dels seus principals beneficis que podem repartir les tasques no només entre diferents processos (*Agents*), sinó també en diferents màquines, provocant així una optimització tant en la codificació com en els recursos.

En el nostre cas, per exemple, tenim l'**Agent Analitzador** especialitzat en mostrar dades tant en format de text com en format gràfic, o l'**Agent Metge** especialitzat en la gestió d'**Agents Pacient** i d'**Alarmes**. A banda que el codi es troba especialitzat per a cada cas, fet que provoca una execució optimitzada, també ens ajuda a descarregar el servidor que es dedicarà majoritàriament a tasques d'accés a la Base de Dades. Els **Agents Pacient** es troben també al servidor ja que fan un ús elevat de les Dades de la Base de Dades, a banda que necessiten estar engegats amb continuïtat per poder processar les auto avaluacions realitzades pels pacients sigui l'hora que sigui.

2.5.3 Fiabilitat

En el nostre cas, degut a les característiques del sistema i la funcionalitat de l'aplicació ens interessa especialment poder comptar amb una sèrie de garanties de **fiabilitat**. En un entorn com en el que ens trobem no ens podem permetre el luxe que el fet que part del sistema deixi de funcionar ens provoqui una caiguda total del sistema.

La **fiabilitat** de la que disposa un *Sistema Multi Agent* ens facilita el control de la situació quan part d'ell pot “caure”, fent que la resta pugui perdre les funcionalitats d'aquesta part, però ens permet la fàcil restauració. A més, en el nostre cas, a no ser que el que pateixi un error de funcionament sigui la part de Serveis, és a dir, la que es trobi en el servidor en el cas de que aquest caigui, no es produirà cap falta de serveis.

2.5.4 Flexibilitat

La **Flexibilitat** que proporciona un *Sistema Multi Agent* ens facilitarà carregar i descarregar *Agents* de forma dinàmica sense gaires complicacions. Això és especialment útil en el cas de que aquesta càrrega/descàrrega s'hagi de realitzar de forma remota com és el cas de la creació i baixa d'**Agents Pacient** que ja es veurà amb detall als capítols següents. Gràcies a la flexibilitat d'un SMA, aquesta càrrega remota acaba resultant gairebé tan senzilla com si es realitzés localment ja que es fa mitjançant la crida a un servei de càrrega i descarrega d'agents que ja ve proporcionat pel *Sistema Multi Agent*.

A banda dels **Agents Pacient**, els **Agents Metge** i l'**Agent Analitzador de Dades** també s'activaran i es desactivaran al llarg del temps degut a l'acció dels usuaris que els engegaran o pararan. Tot això, a més, tenint en compte que molts cops es realitzaran intents de comunicació cap a un agent que no està engegat i que això no provocarà cap problema ja que és de fàcil control.

3 Arquitectura del Sistema

Tal i com s'ha comentat anteriorment, poden diferenciar dos grans blocs en el sistema **PalliaSys**. El primer, format per el sistema basat en *Tecnologies de la informació i les Comunicacions (TIC)* compostat per el projecte del Pablo García Rué [3] més les modificacions que es comentaran posteriorment, com la creació d'un nou perfil d'usuari que ens permetrà l'alta d'usuaris amb el rol de Metge. I el segon, consistent amb el **Sistema Multi Agent (SMA)**, sobre el que s'ha desenvolupat la càrrega principal de treball d'aquest projecte i en el que també s'inclou el projecte del Raul Mateos[2]. Aquests dos grans blocs comparteixen la *Base de Dades* que pot ser considerada com el nexa principal entre ells.

A continuació es passa a explicar més detalladament cada membre d'aquesta i les funcions que realitza. Es començarà fent una ullada a la banda del TIC continuant amb una explicació acurada de la part del *Sistema Multi Agent (SMA)* seguint amb una visió general de la Base de Dades i les taules més importants, per acabar amb un apartat on s'explicaran les mesures de seguretat fetes servir.

3.1 Aplicatiu Web

Com s'ha comentat, aquest bloc està format majoritàriament pel projecte final de carrera del Pablo García Rué [3], per la qual cosa no s'entrarà tan en detall com amb la part del Sistema Multi Agent. Primer es farà una breu introducció a les funcionalitats bàsiques i l'estructura del projecte original per acabar exposant les modificacions realitzades i els motius que han empès a realitzar-les.

3.1.1 Sistema telemàtic per a l'assistència mèdica d'una UCP

El projecte del Pablo García Rué [3] és un *Sistema basat en Tecnologies de la informació i les Comunicacions (TIC)* i avarca totes les necessitats bàsiques de maneigament d'informació tal i com és l'entrada de dades i consultes bàsiques d'aquestes. Està realitzat amb tecnologia web, concretament amb PHP + javascript. Com a servidor s'ha emprat l'Apache HTTP Server amb el mòdul PHP i el mòdul de seguretat SSL.

En el projecte podem diferenciar tres tipus d'usuaris:

- **Pacients:** El pacient és la persona que rep les cures pal·liatives. Pot residir en el seu domicili o estar ingressat en el pavelló de la Unitat de Cures Pal·liatives de l'hospital, en altres pavellons de l'hospital o en un centre soci sanitari. En principi el sistema podrà ser emprat per aquells pacients no hospitalitzats (o persones que se n'encarreguin d'ells en el seu domicili). Les funcions principals d'un pacient seran les següents:
 - Realitzar una auto avaluació.
 - Accés a les dades personals i modificació d'aquestes.

- o Consultar el tractament assignat.
- o Consultar la informació enviada pel doctor.
- **Metges:** Els metges realitzen visites en la seva consulta i passen visita en els pavellons hospitalaris. En principi no es distingiran diferents tipus d'usuaris (metges i infermeres) del sistema en aquesta categoria. Les funcions principals del perfil del metge són:
 - o Alta del pacient.
 - o Consultar avaluacions pendents.
 - o Realitzar nova avaluació.
 - o Consultar història clínica dels pacients.
 - o Complimentar avaluació final.
- **Administrador:** Aquest rol ha estat introduït posteriorment en el projecte que ens ocupa. En el següent apartat, el de Modificacions, es detallen les funcionalitats que se li han assignat.

3.1.2 Modificacions

Per diverses causes va ser necessari realitzar una sèrie de modificacions sobre el projecte del Pablo García Rué [3]. Un dels motius d'uns quants d'aquest canvis va ser que des de l'Hospital de Sant Pau, després de la instal·lació en fase de proves, es van detectar una sèrie de mancances que van haver de ser resoltes. Com a resultat es van realitzar les següents accions:

- *Creació del perfil de l'administració per la creació d'usuaris amb perfil de Metge:* Es va detectar que per a l'alta de nous usuaris amb rol de metge s'havia de realitzar directament sobre la Base de Dades. Al tractar-se d'unes dades bastant dinàmiques, resultava incòmode haver-ho de fer així, és per això que es va optar per a la creació d'un nou rol, el de l'Administrador. La idea d'aquest nou perfil és de que sigui únic, és a dir, que d'administrador només n'hi haurà un, conseqüentment no resulta costós que aquest s'entri directament a la Base de Dades. A canvi, des d'ell podem generar noves comptes amb el perfil de Metge sense haver d'accedir directament a la Base de Dades. A més, aquest perfil també es va aprofitar per a encabir-hi una nova funcionalitat que veurem a continuació.
- *Creació de còpies de seguretat de la Base de Dades de forma senzilla:* De la mateixa manera que era poc pràctic la generació d'usuaris directament des de la Base de Dades, també resultava certament incòmode que per a realitzar les còpies de seguretat de la Base de Dades s'hagués de realitzar in situ, al servidor on es trobés la BD. Per això, i aprofitant l'avinentsa de l'aparició del perfil de l'administrador, es va aprofitar per a encabir-hi també la funcionalitat de còpia de la Base de Dades. D'aquesta forma, des del perfil de l'administrador es poden generar còpies de seguretat de la BD. S'ha de tenir en compte que per cada dia només es pot guardar una còpia, sempre la més actual.

- *Opcionalitat dels camps d'introducció de dades:* En molts casos es requeria que s'omplissin una sèrie de camps als formularis dels que els metges no sempre disposen de les dades al moment d'omplir-los.

A banda d'aquestes modificacions, també va ser necessari introduir-ne una més que era necessària per al projecte que ocupa aquestes pàgines.

- *Comunicació amb el SMA a través de l'obertura d'un socket:* Com veurem més endavant, cada cop que un pacient realitza una Auto Avaluació el Sistema Multi Agent analitzarà les dades que ha entrat el pacient per a veure si ha de fer saltar una Alarma. Conseqüentment es va haver de retocar el codi per tal de realitzar l'obertura del Socket i la posterior transferència d'informació i el tancament d'aquest.

3.2 Sistema Multi Agent

El *Sistema Multi Agent*, al ser distribuït, es divideix en dues parts bastant diferenciades entre elles: la que es troba a la *Plataforma Principal*, al servidor, juntament amb el *Servidor Web* i la *Base de Dades* i la que es troben les màquines dels usuaris. En la **Figura 4** podem veure amb detall com es troba distribuït el SMA i com es relacionen els agents entre ells. En aquest esquema hem obviat els agents que venen per defecte amb la plataforma.

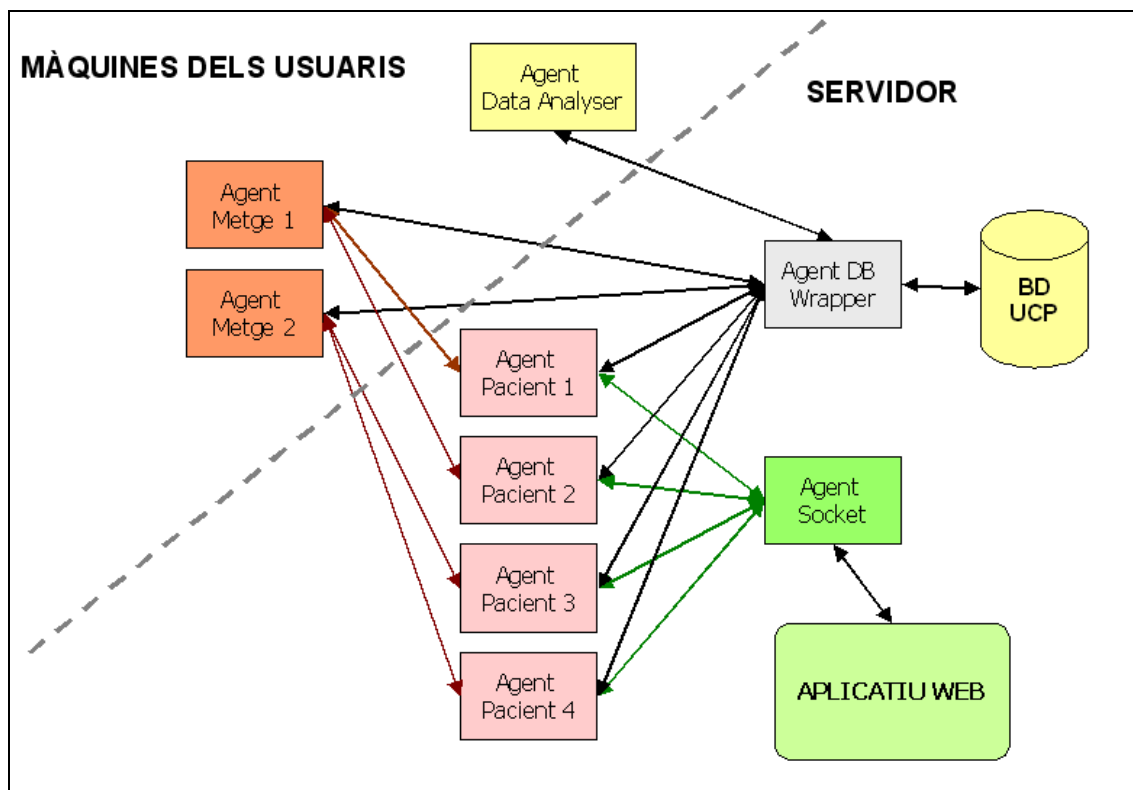


Figura 4 - Arquitectura del SMA

Els agents continguts dintre de la plataforma del servidor es caracteritzen tots per no disposar d'interfície gràfica ja que amb ells no ha d'interactuar cap usuari. Dintre d'aquest grup hi ha l'**Agent Socket**, l'**Agent DBWrapper** i els **Agents Pacient**.

La banda del SMA que es troba a les màquines dels usuaris és la composta per els **Agents Metge** i l'**Agent Analitzador**. Tots aquests, a diferència dels anteriors, disposen d'interfície gràfica ja que la seva funció principal és la d'interactuar amb l'usuari.

A continuació s'explicarà un a un els diferents tipus d'agents que configuren aquest SMA. En primer lloc es parlarà dels agents sense interfície gràfica i es continuarà amb els agents amb interfície. Per cada agent s'explicarà quina és la seva funció, el seu cicle de vida, la interacció amb els altres agents i al final, quins comportaments (*Behaviours*) s'han fet servir. D'aquesta forma veurem més clarament els avantatges del **Jade** i el funcionament de cada *Agent*. Els *behaviours* els explicarem el primer cop que s'anomenin. I si, més tard, un altre agent el fa servir, llavors s'anomenarà de passada, sense entrar és més detalls. Cal recordar que per a fer servir un *Behaviour* el que se sol fer és estendre'l i sobreescriure els mètodes que es necessitin.

També dedicarem un apartat als processos de comunicació existents entre els diferents agents que componen el sistema, on detallarem el protocol de la FIPA que s'ha emprat.

3.2.1 Agent Pacient

Els **Agents Pacient**, com ja hem comentat, són agents sense *interfície gràfica* l'existència dels quals romandria amagada de l'usuari si no fos perquè es creen i eliminen des de l'**Agent Metge**.

La seva funció és la de comprovar si les alarmes que li ha assignat l'**Agent Metge** es compleixen. Aquesta comprovació es portarà a cap cada cop que el pacient que tingui assignat realitzi una auto avaluació. En el cas de que així sigui se n'haurà d'encarregar d'avisar a l'**Agent Metge**. L'existència d'un **Agent Pacient** anirà lligada a l'existència del pacient corresponent. Per cada pacient podrem disposar únicament d'un **Agent Pacient**.

3.2.1.1 Cicle de vida

El cicle de vida d'un **Agent Pacient** passa per les següents fases que comentem a continuació:

- 1) **Creació de l'Agent per part de l'Agent Metge**: L'Agent Metge procedirà a crear l'Agent pacient a la plataforma del Servidor. A l'hora de crear-lo es realitzaran una sèrie de tasques de validació en les que es realitzaran les següents comprovacions:

- a. *Que el pacient existeixi:* Per tal de que ens sigui d'utilitat, un Agent Metge ha de disposar d'un pacient al que estar associat per a realitzar el control de les auto avaluacions.
 - b. *Que l'Agent pacient no estigui creat prèviament:* Tal i com s'ha comentat, un pacient només pot tenir assignat un Agent Pacient.
- 2) **Assignació de les Alarmes per part de l'Agent Metge:** L'Agent metge serà l'encarregat d'assignar les alarmes corresponents a l'Agent Pacient per a que pugui avaluar les diferents auto avaluacions, altrament l'Agent Pacient no podrà realitzar cap tasca, el Aquest punt es comentarà amb més detall a l'apartat de les alarmes.
 - 3) **Esperar un avís de nova Auto avaluació:** Mentre un Agent Pacient està actiu està en un estat constant d'espera. Pot esperar rebre un avís per part de l'Agent Socket de que s'hagi produït una nova Auto avaluació per part del Pacient o també pot esperar una modificació de les alarmes que té assignades per part de l'Agent Metge.
 - 4) **Tractament i anàlisi d'una auto avaluació:** En el cas de que rebí un avís per part de l'Agent Socket, analitzar les alarmes que té assignades. Si una alarma es valida, llavors haurà d'avisar a l'Agent Metge de l'activació de l'alarma.
 - 5) **Baixa de l'Agent Pacient:** El seu cicle de vida finalitza quan l'Agent Metge el dona de Baixa. Això esborrarà qualsevol rastre d'aquest al sistema i ho deixarà tot llest perquè aquest o qualsevol altre Agent Metge el torni a donar d'alta en el cas de que sigui necessari.

Tal i com acabem de veure un **Agent Pacient** és un element que es crearà i eliminarà de forma dinàmica un cop el sistema ja estigui posat en funcionament. En aquest cas, com ja s'ha comentat amb anterioritat, la flexibilitat proporcionada pel SMA per tal de carregar i descarregar dinàmicament els agents del sistema simplifica aquesta tasca. Aquesta càrrega i descàrrega es realitzarà mitjançant els serveis que ens ofereix l'Agent

3.2.1.2 Behaviours

L'**Agent Pacient** és l'agent que es comunica amb més tipus d'Agents després de l'**Agent DBWrapper**, i per cada acte comunicatiu disposarà d'un comportament particular. És per tot això que fa servir més d'un tipus de *Behaviour*. A continuació els llistem i fem una breu explicació de quin és el seu funcionament, cicle de vida i la tasca que realitzen:

- **Esperar Auto avaluació (AchieveREResponder):** Aquest comportament s'engegarà quan l'Agent s'arrenqui manenint-se en funcionament al llarg de tota la vida de l'Agent. Serà el que s'encarregarà d'esperar i tractar les peticions per

part de l'**Agent Socket** quan l'avis de que el pacient ha realitzat una nova auto avaluació.

Aquest comportament és cíclic i cada cop que tracti una petició es reiniciarà i esperarà a rebre'n una de nova. El *behaviour* simètric que trobarem a la banda de l'Agent Socket serà l'*AchieveREInitiator*.

- **Esperar subscripció d'Alarmes (*SubscriptionResponder*):** A l'igual que en cas anterior, s'engegarà juntament amb l'agent i també es mantindrà en funcionament mentre l'agent estigui actiu. Aquest serà el *behaviour* encarregat de la gestió de les alarmes.

Abans s'ha comentat que l'**Agent Metge** assigna les alarmes a l'Agent Pacient, però no s'ha comentat com ho feia. Ho fa mitjançant un protocol de subscripció, ja que es podria dir que l'Agent Pacient proporciona el servei de validació d'alarmes al que es pot subscriure l'Agent Metge. Aquest és pràctic ja que amb aquest sistema les coses queden més automatitzades, a més, si en un futur es vol fer que un Agent Pacient pugui estar associat a més d'un Agent Metge, el fet d'haver-se plantejat d'aquesta forma el procés d'assignació d'Alarmes, facilita les coses.

En l'apartat de comunicació entre agents que es troba al finalitzar l'explicació dels agents s'explicarà com funciona el protocol de subscripció.

- **Consulta a la Base de Dades (*AchieveREInitiator*):** Aquest comportament es farà servir cada cop que es vulgui realitzar una petició a la base de Dades per tal d'obtenir les dades de les auto avaluacions. En aquest cas, l'execució d'aquest *behaviour* s'inicia quan es volen obtenir les dades i finalitza quan es reben. Envia un petició a l'**Agent DBWrapper** i processa la resposta.

3.2.2 Agent Socket

L'**Agent Socket** també és un agent sense interfície gràfica i aquest, a l'igual que l'**Agent DBWrapper** que es veurà a continuació, sí que és invisible per als usuaris. La seva funció és a d'actuar com a enllaç entre la part Web del projecte i el *Sistema Multi Agent*. Disposa d'un **Socket** obert a l'espera de connexions per la banda de l'aplicació web, les quals anirà tractant en ordre de rebuda. Aquestes connexions són les que informen de que un determinat pacient acaba de realitzar una auto avaluació.

Quan l'Agent Socket rebí una connexió per part de la banda de l'aplicatiu web, haurà de procedir a avisar a l'**Agent Pacient** corresponent, és a dir, el que estigui associat a el pacient que ha realitzat la auto avaluació. És possible que a un pacient no tingui associat cap **Agent Pacient**.

La comunicació a través del **Socket** es realitzarà de forma xifrada per tal de garantir al màxim la seguretat i la privacitat de les dades. Com es veurà més endavant, aquest és un punt important.

A diferència de l'Agent Pacient, l'Agent Socket s'activarà al moment en que engegarem el sistema i es mantindrà en funcionament sempre. Una altra característica d'aquest agent és que a diferència de la resta, no accedeix a la *Base de Dades*.

3.2.2.1 Behaviours

En el cas d'aquest agent disposarem dos *Behaviours*, un que es farà servir per a comunicar-se amb altres agents del sistema (Agents Pacient) i un altre que es comunicarà amb la part web.

- **Esperar Connexió (CyclicBehaviour)**: Aquest comportament es posarà en funcionament quan l'Agent Socket s'engegui i es mantindrà actiu al llarg de tota la vida de l'agent.

Serà l'encarregat d'esperar les connexions que puguin arribar des de la part de l'*aplicatiu web*. Tal i com indica el seu nom, és cíclic, és a dir, que es reiniciarà cada cop ell sol. A diferència dels comportaments que hem descrit fins ara, és el primer que no disposa d'un *Behaviour* simètric ja que no realitza cap tasca de comunicació entre dos agents del sistema..

- **Informar sobre realització Auto avaluació (AchieveREInitiator)**: Quan rebem una connexió des de la banda de l'*aplicatiu web*, engegarem aquest comportament per tal d'avisar a l'**Agent Pacient** corresponent, en el cas que aquest existeixi, de que el pacient al que està associat acaba de realitzar una auto avaluació i a ell li toca comprovar les alarmes i avisar a l'**Agent Metge** si és necessari.

3.2.3 Agent DBWrapper

És l'encarregat de gestionar l'accés a la *Base de Dades* (BD) per la banda del **SMA**, funciona com una interfície amb la *Base de Dades* interactuant amb aquesta i l'exterior, tenint la funció principal de servir les peticions d'inserció, modificació, eliminació i consulta de les dades referents als pacients, per part dels agents. El fet de disposar de l'Agent DBWrapper que ens realitza aquesta funció de control d'accés a la Base de Dades ens proporciona més seguretat i sincronització en l'accés.

Per tal de comunicar-se amb ell, la resta d'agents del sistema, hauran de fer-ho mitjançant un protocol de la **FIPA** [4], concretament el *FIPA Request Protocol*. El tractament de les peticions és relativament senzill: cada cop que l'Agent DBWrapper rebí una petició de consulta o modificació de la Base de Dades, aquesta s'encuarà en una cua FIFO (*First In First Out*) i es tractarà després de que les que hagin arribat abans hagin estat tractades. Per motius de seguretat, l'Agent DBWrapper solament acceptarà peticions que tinguí ell definides. D'aquesta forma es limita l'accés a les dades estrictament necessàries. A més, l'agent DBWrapper també exerceix el paper de

controlador d'accés a les dades del sistema demanant autenticació a tot aquell que intenti connectar amb la Base de Dades.

Per altra banda, les comunicacions entre l'Agent DBWrapper i els altres agents es realitzen de forma xifrada mitjançant SSL, necessària per la transmissió de dades crítiques mitjançant canals de comunicació de caràcter obert.

Com es comentarà més endavant, l'Agent DBWrapper també és l'encarregat de gestionar els **fitxers de log**. Això és degut a que és ell qui centralitza i acapara l'accés a la *Base de Dades* en el *Sistema Multi Agent*. Com s'ha comentat abans, demana autenticació a tot aquell que s'intenti connectar a la Base de Dades, un cop obtinguda aquesta autenticació procedirà a emmagatzemar l'informació al fitxer de log corresponent.

A l'igual que l'Agent Socket, aquest agent s'engegarà al mateix moment en que es posi en funcionament el sistema i es mantindrà actiu mentre aquest funcioni.

3.2.3.1 Behaviours

L'Agent DBWrapper, degut a la seva especialització, disposa d'un únic *behaviour*, el **CyclicBehaviour**. Aquest serà l'encarregat d'anar tractant les peticions d'accés a la Base de Dades per la resta dels Agents del sistema. Amb la petició de consulta o modificació a la Base de Dades, el que arriba és un identificador amb el nom de la consulta a realitzar i els paràmetres que es faran servir.

3.2.4 Agent Metge

L'**Agent Metge** possiblement és l'agent més significatiu del sistema ja que, a l'igual que l'**Agent Analitzador**, disposa d'interfície gràfica que permetrà a l'usuari interactuar amb ell directament. I a més, és des d'ell d'on es realitza la gestió dels **Agents Pacient**, la generació i assignació **d'Alarmes** i la recepció d'avisos de l'activació de les **Alarmes**.

Cada metge disposarà del seu respectiu **Agent Metge** instal·lat en el seu ordinador personal. Per tal d'obtenir les dades necessàries, així com per emmagatzemar les diferents alarmes que vagi definint, necessitarà comunicar-se de forma remota amb l'**Agent DBWrapper** que es trobarà a la plataforma d'agents que estarà corrent al servidor. Per a fer-ho, farà ús del *FIPA Request Protocol* definit per la FIPA [4].

Com ja hem comentat més amunt, les tasques principals dels Agents Metges són la gestió de pacients i Alarmes. Seràn ells els encarregats de tot el procés de creació i manteniment.

Per a la creació i assignació d'alarmes, que explicarem més detalladament en la secció d'Alarmes, així com per a l'eliminació d'agents pacient, necessitarà fer-ho de forma remota ja que tots els agents pacients es trobaran corrent a la plataforma remota que estarà al Servidor. Per a la Baixa i Alta d'agents pacient es comunicarà amb l'**AMS**.

3.2.4.1 Behaviours

L'Agent Metge disposa de tres *behaviours* per a comunicar-se amb la resta d'Agents del sistema. En el seu cas, degut a les seves funcions, necessitarà comunicar-se amb l'**Agent DBWrapper** i els **Agents Pacient** que tingui assignats. A continuació explicarem quins son aquests dos comportaments:

- **Assignació d'Alarmes (*SubscriptionInitiator*)**: Aquest serà el comportament que farem servir quan vulguem assignar una alarma a un Agent Pacient.

Aquest comportament es posarà en funcionament en el moment de fer l'assignació i serà l'encarregat de rebre les notificacions per part de l'Agent Pacient al que s'hagi subscrit. Aquest comportament és simètric al *SubscriptionREInitiator*.

- **Consulta a la Base de Dades (*AchieveREInitiator*)**: Com en els altres casos de consulta a la Base de Dades, aquest *behaviour* es comportarà de la mateixa forma, engegant-se quan es requereixin dades de la Base de Dades.
- **Consulta Múltiple a la Base de Dades (*SequentialBehaviour*)**: En aquest agent disposem de consultes que poden tenir una certa complexitat, és per això que és necessari aquest comportament. Com els dos anteriors, s'engegarà quan sigui requerit per l'agent.

La seva funció serà la de generar **subBehaviours**, és a dir, *behaviours* dependents d'ell. Això és útil quan necessitem esperar a que finalitzin el seu funcionament una sèrie de comportaments per a poder dur a terme una tasca. Així doncs, el *SequentialBehaviour* esperarà a que tots els seus fills finalitzin l'execució per a acabar ell.

Tal i com hem vist, degut a la característica de ser un Agent gràfic que actua quan l'usuari li demana, no tenim cap comportament que s'engegui quan l'agent es posa en funcionament i es mantingui actiu fins que aquest finalitza.

3.2.5 Agent Analitzador

Tal i com diu el seu nom, és l'Agent que analitzarà i presentarà les dades contingudes a la *Base de Dades*. Des de l'**Agent Analitzador** podrem generar estadístiques i gràfiques amb les dades que s'hagin recollit. Ens proporcionarà bastanta flexibilitat a l'hora de definir com son les estadístiques a generar ja que ens permetrà escollir períodes i també grups de dades a mostrar en l'informe.

A banda, també ens permetrà la definició d'*Alarmes* que posteriorment podran fer servir els Agents Metge per a assignar als seus Agents Pacient. Aquesta opció de definició d'*Alarmes Globals* és característica d'aquest agent.

3.2.5.1 Behaviours

Degut a les seves funcions, l'**Agent Analitzador** només disposarà de dos tipus de Behaviours i tots per a comunicar-se amb l'**Agent DBWrapper**.

- **Consulta a la Base de Dades (AchieveREInitiator)**: A l'igual que en els altres agents aquest behaviour de consulta a la Base de Dades, aquest *behaviour* es comportarà de la mateixa forma, engegant-se quan es requereixin dades de la Base de Dades.
- **Consulta Múltiple a la Base de Dades (SequentialBehaviour)**: Per a obtenir segons quines dades per les estadístiques se'ns generen consultes que poden tenir una certa complexitat, és per això que és necessari aquest comportament. Com l'anterior, s'engegarà quan sigui requerit per l'agent.

3.2.6 Comunicació entre Agents

A l'hora d'explicar els agents s'ha parlat dels comportaments (*behaviours*) dels que disposen, i això, juntament amb els comportaments i els cicles de vida dels agents, ja dona una idea de quins seran els processos de comunicació que s'han fet servir. Parlarem dels dos més importants: el de Subscripció d'alarmes per part de l'agent Metge i un de genèric de comunicació amb la *Base de Dades*, és a dir, amb l'**Agent DBWrapper**.

Aquests processos de comunicació fan ús de protocols d'interacció que estan compostats per dos *Behaviours* simètrics, un *Initiator* a la banda de l'Agent que inicia l'acte comunicatiu i un *Responder* a la banda de l'altre Agent, anomenat a vegades com a *Participant*.

3.2.6.1 Subscripció d'Alarmes (FIPA-Subscribe)

Com ja s'ha comentat, per la subscripció d'alarmes farem ús del **FIPA-Subscribe**. Aquest protocol d'interacció permet a l'*Initiator* enviar un missatge de subscripció al *Participant* indicant quina és la subscripció desitjada, en el nostre cas es tractarà de la subscripció a la validació de l'**Alarma** que nosaltres li indicarem. El *Participant* processa el missatge de subscripció i respon a la demanda acceptant o rebutjant la subscripció. Si el *Participant* refusa la petició, respon amb un **refuse**, altrament, si el *Participant* hi està d'acord, pot comunicar un missatge opcional d'acceptació.

Si el *Participant* ha acceptat la subscripció, cada cop que el que se'ns ha demanat es compleixi, en el nostre cas, que una de les alarmes doni positiu en la validació realitzada, ho comunica a l'Agent que ha realitzat la subscripció mitjançant un **inform-result**. El *Participant* continuarà enviant informació sobre uns subscripció fins que l'*Initiator* cancel·li la subscripció amb un missatge d'error, o el *Participant* experimenti una fallada, comunicat amb un **missatge de fallada**.

Podem veure la representació del protocol d'interacció a la **Figura 5**.

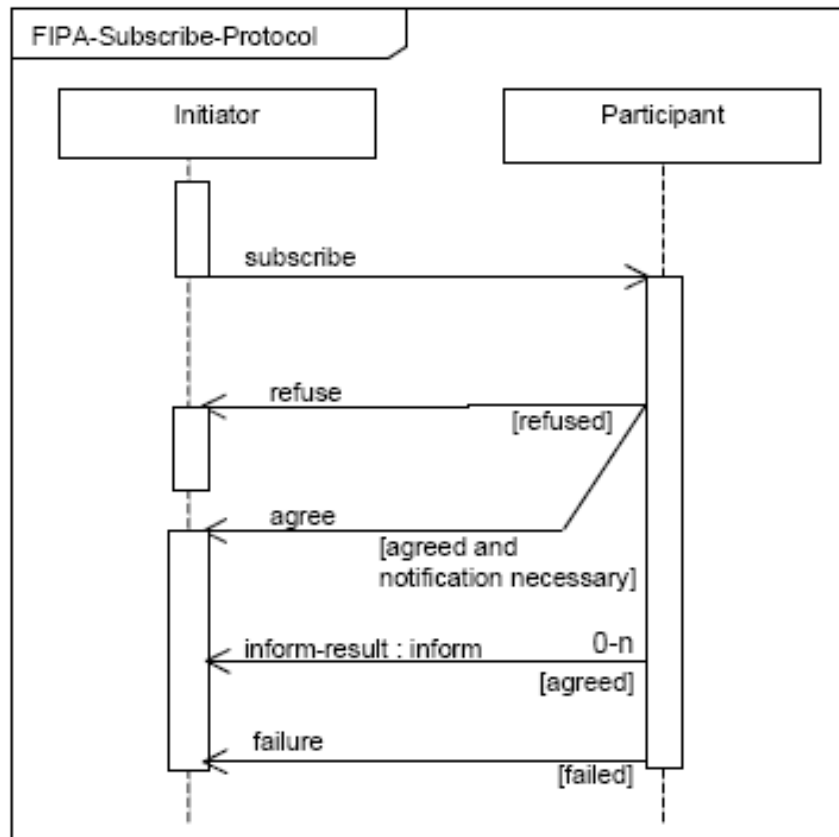


Figura 5 - Esquema del protocol de Subscripció

3.2.6.2 Petició a la Base de Dades (*FIPA Request Protocol*)

Per a la petició d'informació a la Base de Dades en qualsevol dels Agents, es fa servir el **FIPA Request Protocol [9]**. L'*Initiator* iniciarà l'acte de comunicació enviant una petició al *Responder*, en el nostre cas, serà una acció a realitzar sobre la Base de Dades que pot ser una lectura o una modificació. Un cop rebí el missatge, el *Responder* el processarà i ens contestarà o amb un **not-understood** (en el cas que no hagi entès la nostra sol·licitud), un **refuse** (en el cas que no accepti la nostra petició) o amb un **agree** (en el cas de que l'accepti). Si accepta la petició de l'*Initiator*, passarà a processar-la per a, després, enviar-li un **missatge de fallida** (en el cas que no hagi pogut portar a terme l'acció amb èxit) o un **inform** en el cas que tot hagi anat correctament. En aquest missatge, en el nostre cas, si és una petició d'informació, serà per on ens arribarà la informació.

En la **Figura 6**, podem veure un diagrama de com es porta a terme aquest acte comunicatiu.

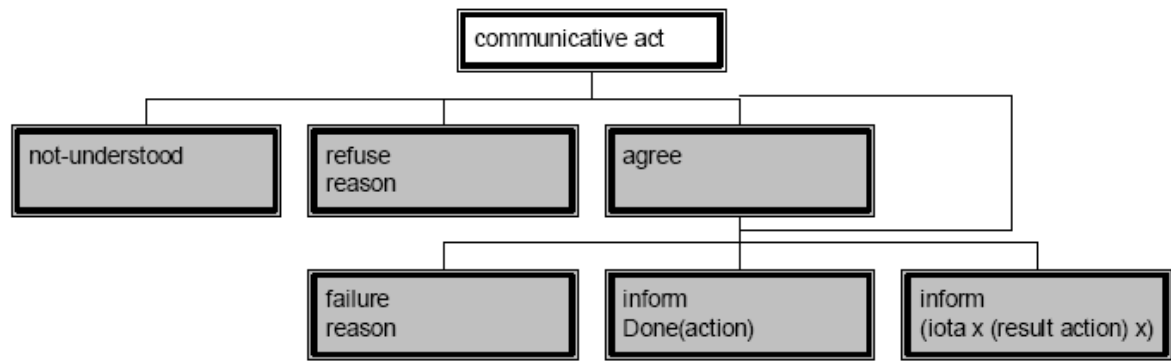


Figura 6 - Esquema del Request protocol

3.3 Base de Dades

En aquest apartat entrarem en detall en el disseny i implementació de la **Base de Dades**. En primer lloc farem una breu explicació dels passos que s'han seguit per al disseny per a continuar amb la presentació del model relacional de la mateixa i acabar parlant del contingut de les principals taules i la seva funció. Degut a que la base de dades està compartida per els dos blocs principals de *Palliasys* no explicarem totes les taules, sinó que només entrarem en detall en les que es fan servir en aquest projecte.

3.3.1 Introducció

Per al disseny de la **Base de Dades** ha estat necessari realitzar prèviament un estudi de les necessitat, el funcionament de la *Unitat de Cures Pal·liatives* i la forma en que es porta l'assistència mèdica de la unitat. També s'ha estudiat quina era la informació que recollien els metges dels pacients, així com es tractava aquesta.

La informació dels pacients era recollida en informes de paper en els que, en primer lloc, es guardava l'alta d'un pacient amb totes les seves dades personals, episodi, diagnòstic i estat de salut. A partir d'aquí, i a través de les successives visites amb els pacients, aquests anaven omplint la fulla amb els factors d'auto avaluació per a que posteriorment els metges els complementessin amb les seves valoracions.

Després d'estudiar tots els informes i entendre la forma de treballar dels metges de la *Unitat de Cures Pal·liatives*, es van discutir noves funcionalitats que es podrien afegir en el sistema a desenvolupar. Per a anomenar alguna de les novetats, es va decidir mantenir informació dels canvis d'ubicació dels pacients des del moment en el que ingressen a la unitat.

Una vegada es va comprendre el funcionament que havia de tenir el sistema, es va passar a dissenyar la **Base de Dades relacional** en la que s'hauria d'emmagatzemar tota la informació dels pacients i dels metges. Es va decidir implementar-la en *Microsoft Access* sobretot perquè altres unitats de l'Hospital feien ús d'aquesta **Base de Dades** i així era més senzilla una futura integració de sistemes. Per a accedir a al Base de dades dels del servidor Web, es fa a través del **ODBC**. I per accedir-hi a través del Sistema

Multi Agent, tal i com s'ha comentat anteriorment, es fa des del **DBWrapper**. Aquest fa servir una **connexió JDBC**.

A banda del procés de disseny que s'ha explicat, la **Base de Dades** ha patit una remodelació per tal d'adaptar-la als canvis realitzats sobre el projecte del Pablo García Rué [3] i també per a introduir-hi noves taules que eren necessàries en aquest projecte, com les referents a les **Alarmes**.

3.3.2 Model Relacional

Per a poder entendre millor la relació entre les taules i les decisions preses, primer de tot podem veure el model relacional en la **Figura 7**.

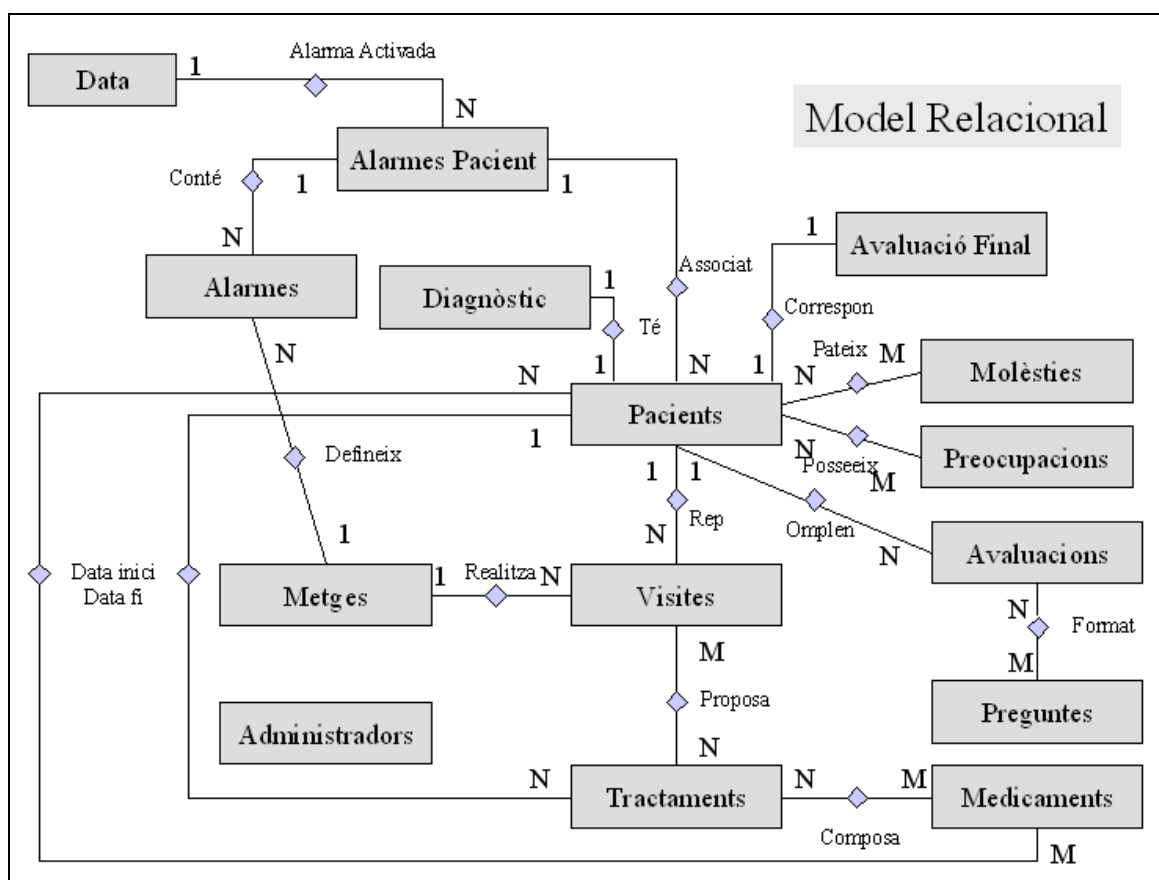


Figura 7 - Model Relacional de la Base de Dades

3.3.3 Taules

Cada taula de la **Base de Dades** té la seva tasca i conté informació clau per a l'aplicació. Com s'ha comentat, ara es procedirà a analitzar les taules més importants i significatives per al *Sistema Multi Agent*, es realitzarà una breu descripció i es ressaltaran els camps de major importància. Els camps que apareixen subratllats són els que corresponen a la clau primària.

3.3.3.1 Taules dels usuaris del sistema

En aquestes taules és on trobarem la informació bàsica dels usuaris relacionada amb el sistema, com són les dades necessàries per a l'autenticació amb l'accés. També és on trobarem les dades personals que no s'acostumen a modificar.

- **Pacients:** Es la taula principal de la Base de Dades. Conté les dades personals dels pacients, entre ells el número d'història clínica i el password per a accedir a l'aplicació des de l'aplicació web. Aquesta informació s'omple quan el pacient s'incorpora a la unitat i rarament es modifica amb excepció del password.

Camps: Número d'història clínica (NHC), primer cognom, segon cognom, nom, data de naixement, sexe, direcció, població, codi postal, telèfon, password.

- **Metges:** Conté la informació sobre els metges necessària per a que puguin accedir al sistema. A l'igual que la taula anterior, l'emmagatzemament d'aquestes dades es realitza quan el metge es donat d'alta i la seva modificació no és habitual, exceptuant el password

Camps: Identificador del metge, password, nom, primer cognom, segon cognom.

- **Administradors:** En ella es troba la informació referents als administradors necessària per a que puguin accedir al sistema. Es recorda que la figura de l'administrador és la que gestiona als metges dintre de l'aplicació.

Camps: Identificador de l'administrador, password, nom, primer cognom, segon cognom.

3.3.3.2 Taules relacionades amb les Alarmes

Tota la informació relacionada amb les Alarmes la trobarem en aquestes taules. En elles tindrem guardada tota la informació referent al cicle de vida d'aquestes, des de la definició de l'equació, l'assignació i l'activació.

- **Alarmes:** Quan un Agent Metge o l'Agent Administrador defineixi una alarma, aquesta taula és on es guardarà. Hi ha més d'un tipus d'alarma tal i com veurem a la secció d'alarmes. Però totes es guarden a la mateixa taula.

Camps: Nom de l'Alarma, Identificador del metge propietari de l'Alarma, Equació, Tipus d'Alarma (**Bàsica**, **Evolució**).

- **Alarmes Pacient:** Aquí és on es guardarà l'associació de les alarmes amb els diferents Agent Pacient. Un pacient pot tenir més d'una alarma associada i les

alarmes poden estar associades a més d'un pacient, però una alarma no pot estar assignada més d'un cop a un mateix pacient.

Camps: Identificador del pacient al que associem l'Alarma (NHC), Identificador del Metge propietari de l'Alarma, Nom de l'Alarma.

- **Alarmes Activades:** Quan una alarma s'activa, a banda d'avisar a l'Agent Metge corresponent si està activat, també es procedirà a guardar un registre en aquesta taula. Quan l'Agent metge hagi vist l'Alarma i pres les mesures adequades, podrà decidir passar-la a històrica.

Camps: Identificador del pacient al que associem l'Alarma (NHC), Identificador del Metge propietari de l'Alarma, Nom de l'Alarma, Data d'Activació, Indicador d'Històric.

3.3.3.3 Taules relacionades amb les dades mèdiques dels pacients

En aquestes taules és on s'emmagatzemaran les dades mèdiques dels pacients. En elles podem trobar des del diagnòstic inicial que se'ls hi va realitzar a l'entrar a la Unitat de Cures Pal·liatives de l'Hospital fins a l'Historial de les auto avaluacions.

- **Avaluacions:** És on s'aniran guardant l'historial d'auto avaluacions dels pacients. Aquesta és la taula de la que es s'extraurà la informació per tal de comprovar si les alarmes s'activen.

Camps: Identificador del pacient (NHC), Número d'assistència, data de la avaluació, persona que informa, ubicació del pacient i tots els camps de l'avaluació (grau de dolor, nàusees, insomni, depressió, etc.).

- **Avaluació Final:** En aquesta taula guardem les últimes dades relacionades amb el pacient. Tant les que fan referència al seu estat de salut final, valoracions globals del tractament, així com l'estat psicològic de la seva família i del mateix pacient.

Camps: Identificador del pacient (NHC), data d'Exitus, temps en el programa, temps en el domicili, temps amb els PADES, seguiment en consultes externes (Sí, No), Valoració global dels símptomes, problemes no ben controlats, opioïde emprat, dades de l'estat global final del pacient i la seva família (petició d'eutanàsia, serenitat familiar, angixa terminal, etc.).

- **Diagnòstics:** Aquí hi trobarem les dades referents a quan el pacient va ser ingressat a la Unitat de Cures Pal·liatives, els tractaments previs, el pronòstic, etc. Aquesta taula està complementada amb d'altres taules dependents que ara no comentarem, però que complementen la informació continguda en aquesta.

Camps: Identificador del pacient (NHC), Càncer, codi del càncer, localització, estadi del càncer, metàstasi, tipus de metàstasi (òsea, hepàtica, etc.), tractaments previs, cirurgia realitzada, quimioteràpia, tipus de quimioteràpia, altres, malalties associades, motius de la consulta (dolor, debilitat, nàusees, somnolència, anorexia, etc.), claudicació familiar, altres.

3.4 Seguretat

Un apartat bastant important d'aquest projecte ha estat la seguretat. El sistema treballa amb informació confidencial de pacients que requereix una sèrie de mesures de protecció de dades especial. Entre l'any 2000 i 2002, en l'*Hospital de Sant Pau i la Santa Creu* es va realitzar la implantació progressiva d'una sèrie d'objectius referents a mesures de seguretat en els sistemes d'informació existents. Aquests criteris es troben definits en el **Real Decret 994/1999** de l'11 de Juny. Els nous sistemes que s'implantin a partir d'aquell moment necessiten seguir aquests requeriments de seguretat.

En el següent apartat es detallaran un per un aquests criteris, evidentment només es comentaran els que afectin directament aquest aplicació, i posteriorment es descriurà quines son les solucions que s'han emprat per a implementar-los.

3.4.1 Requeriments de seguretat

Els requeriments de seguretat els podem dividir entre els que tindran un efecte directe en l'aplicació i els que hauran de ser portats a terme per el personal de la *Unitat de Cures Pal·liatives de l'Hospital de Sant Pau*.

3.4.1.1 Requeriments que afecten directament a l'aplicatiu

Aquests requeriments seran responsabilitat del sistema *Palliasys*. És a dir, hauran d'estar implementats i duts a terme per el mateix sense la necessitat de l'intervenció dels usuaris, en aquest cas el personal que se n'encarrega de la seguretat en l'*Hospital de Sant Pau i la Santa Creu*. L'únic contacte que ha de tenir el personal de seguretat és el d'imprimir els resultats dels fitxers de *log* que es puguin generar o la generació d'algun que d'altre llistat com el d'intents d'accés fallits.

- **Perfils d'usuari amb diferents permisos d'accés i modificació de les dades:** No totes les persones que accedeixin a l'aplicació necessiten realitzar les mateixes tasques ni tampoc estan autoritzats a poder accedir a totes les dades. Depenent del perfil tindran accés a un determinat sector de l'aplicació que també implicarà l'accés a un determinat grup de dades amb els corresponents permisos de modificació o consulta. Els perfils principals són el d'*Administrador*, el de *Metge* i el de *Pacient*.
- **Registre de tots els intents d'accés a l'aplicació tant els correctes com els fallits:** Serà necessari registrar qualsevol intent d'accés a la informació, tant si és incorrecte com si acaba a bon terme. Les dades que s'hauran d'emmagatzemar seran l'hora, el dia i la identificació que s'ha emprat per a tal propòsit. La

identificació estarà formada per *identificador* i *password* (en el cas d'accésos incorrectes).

- **Registre dels accessos i modificacions a l'historial i informació personal dels pacients:** A banda dels intents d'accés, si aquests són correctes, també serà necessari portar un registre de la informació que ha estat accedida o modificada. Les dades a guardar seran similars a les del punt anterior, amb l'afegit de les dades a les que s'ha accedit.
- **Les comunicacions per xarxa on viatgi informació dels pacients han de ser xifrades:** Amb aquest punt es fa referència a que les dades que viatgin per la xarxa (recordem que ens trobem davant d'un sistema distribuït), han de complir una sèrie de principis: el d'*Integritat*, el de *Confidencialitat*, el d'*Autenticació* i el de *No repudi*.
- **L'administrador ha de poder fer llistats dels accessos incorrectes:** És una forma de tenir controlats amb més eficiència els accessos incorrectes i poder controlar millor els atacs malintencionats.

3.4.1.2 Requeriments de responsabilitat per part del personal de la UCP

Aquests requeriments seran responsabilitat del personal de la Unitat de Cures Pal·liatives. És a dir, no afecten per res la implementació del projecte.

- **Realització de còpies de seguretat dels fitxers de log** (on es registren els diversos intents d'accés per part dels usuaris i el que han visitat i modificat): Periòdicament s'hauran de realitzar còpies de seguretat dels fitxers de *log* que s'hagin generat amb els intents d'accés al sistema i a les dades. S'haurà de guardar un historial d'aquestes còpies fins a quatre anys. Això ho portarà a terme els tècnics d'informàtica de l'Hospital de Sant Pau i la Santa Creu a l'igual que es realitzen les còpies de seguretat d'altres dades. Aquestes còpies de seguretat s'hauran d'emmagatzemar en màquines diferents.
- **Realització d'un document que el pacient firmi conforme les dades s'usaran per a la investigació:** Degut a la normativa existent de protecció de dades, és necessari informar als pacients de que les seves dades s'emmagatzemaran i es faran servir per a analitzar-les i estudiar-les. Això es solucionarà des de la Unitat de Cures Pal·liatives mitjançant la entrega d'un document que el pacient haurà de firmar en el moment del seu ingrés en la Unitat.

Es dona per suposat que el personal d'informàtica de l'hospital de Sant Pau i la Santa Creu s'encarregarà d'aquests dos punts.

3.4.2 Mesures implementades

Per als requeriments que son responsabilitat de l'aplicació, s'han pres diferents mesures. En aquest document parlarem sobre les que s'han pres en el projecte que ens ocupa i explicarem com funcionen.

3.4.2.1 Perfils d'usuari i Autenticació

Com ja s'ha esmentat amb anterioritat, disposem de diferents perfils d'usuaris, concretament de tres tipus de perfils: per a **pacients**, per a **metges** i per a l'**administrador**. Depenent del perfil es pot tenir accés a unes dades o a unes altres. En el projecte que es descriu en aquest document, els rols que tindran veritable importància seran el del Metge i el de l'Administrador ja que els pacients no faran un ús directe del *Sistema Multi Agent*. Per a la correcta comprovació de que l'usuari sigui el correcte, es disposarà d'una validació a l'hora d'entrar en l'**Agent Metge** o l'**Agent Analitzador**. En el primer cas, es farà ús del *perfil de Metge* mentre que en el segon podem entrar tant emprant el *perfil de Metge* com el de l'*Administrador*. Depenent del que fem servir disposarem de més o menys utilitats. A l'hora d'Arrancar l'Agent Metge o l'Agent Analitzador ens apareixerà una pantalla on ens proporcionarà l'identificador i ens requerirà el password. A la **Figura 8** podem veure un exemple de la pantalla de validació.



Figura 8 - Seguratat, pantalla de Validació

3.4.2.2 Registres d'Accés: Fitxers de log

Tant el registre d'intents d'accés (fallits i correctes), com el d'accés a les dades i el llistat d'accessos incorrectes s'han solucionat mitjançant fitxers de *log*. Aquests fitxers de *log* seran generats per l'**Agent Wrapper de Base de Dades** ja que aquest agent centralitza tot l'accés a la Base de Dades del *Sistema Multi Agent* i seran comuns tant per l'**Agent Metge** com per a l'**Agent Analitzador**. Disposarem de tres fitxers de *log* diferents depenent de les dades que hi guardarem:

- **acces.log**: En aquest fitxer s'hi emmagatzemaran tots els intents d'accés que s'han produït tant els fallits com els correctes. El format d'aquest fitxer serà un llistat de línies ordenades per data amb el següent contingut:

AAAA-MM-DD hh:mm:ss Acc Denegat/ AccAcceptat login [password]

Per exemple, es podria trobar el següent:

```
2006-06-21 10:23:09 ACC DENEGAT aquinyones pompa45
2006-06-21 10:23:23 ACC ACCEPTAT aquinyones
2006-06-21 10:26:07 ACC DENEGAT mmateus allevfe
2006-06-21 12:23:27 ACC DENEGAT avadia 456789ty
```

2006-06-22 08:48:09 ACC ACCEPTAT enrics

- **accesErrorni.log**: En aquest fitxer es guarden els intents d'accés fallits. Farà els servei de llistat d'intents d'accés erroris que necessita generar l'administrador. El seu format serà molt similar al del *acces.log*:

AAAA-MM-DD hh:mm:ss login [password]

Un exemple del que ens hi podríem trobar:

2006-06-21 10:23:09 aquinyones pompa45

2006-06-21 10:26:07 mmateus allevfe

2006-06-21 12:23:27 avadia 456789ty

- **dadesConsultades.log**: En aquest fitxer serà on es trobaran els accessos a les dades que hagin pogut realitzar els diferents usuaris. El seu format serà el següent:

AAAA-MM-DD hh:mm:ss login lectura/escriptura taulaAccedida

En aquest fitxer podríem tenir, per exemple:

2006-06-21 10:23:09 aquinyones lectura EVAL

2006-06-21 10:23:23 aquinyones escriptura Alarmes

2006-06-21 10:26:07 mmateus lectura Alarmes

2006-06-21 12:23:27 avadia escriptura Pacients

2006-06-22 08:48:09 enrics escriptura Alarmes

3.4.2.3 Comunicació Segura

En aquest apartat tenim dos casos, el de la comunicació entre l'Aplicació Web i el Sistema Multi Agent i la comunicació dintre del Sistema Multi Agent.

En el primer cas la comunicació es realitzarà a través d'un Socket Segur.

En el cas de la comunicació interna s'ha emprat el **JADE-S** (La versió 2). Aquest és un mòdul de Jade que ens proporciona la capacitat de xifrar la informació que enviem a través de la xarxa.

Per tal d'aconseguir aquest propòsit fa ús de la *Signatura Digital* i l'*encriptació*. De tots és sabut que les signatures serveixen per a assegurar l'**integritat** dels missatges, és a dir, que les dades no han estat modificades durant el procés de transmissió i que la identitat del qui ha generat el missatge és qui diu ser. Per l'altra banda, l'encriptació ens assegura la **confidencialitat** del missatge protegint les dades dels que no estan autoritzat a veure-les. És a dir, ens assegura que solament el receptor a qui anava destinat el missatge sigui capaç de rebre i llegir el missatge amb claredat. Com a informació de rerafons, un missatge ACL està compost per dues part: l'embolcall, que conté la informació relacionada amb el transport i en segon lloc la secció amb les dades.

En el **JADE-S**, la Signatura i la encriptació són aplicades a tota la secció de dades en ordre de protegir totes les peces d'informació contingudes en els slots del missatge ACL (contingut, protocol, ontologia, etc.) La informació relacionada amb la seguretat (tal com la signatura, l'algorisme o la clau s'emmagatzema dintre de l'embolcall. L'usuari no ha de treballar directament amb els mecanismes de signatura i encriptació, solament haurà de requerir que el missatge sigui signat o comprovar que el missatge que ha rebut està signat.

Si es produeix algun problema durant el procés de signatura, encriptació, verificació o desencriptació, el missatge es descarta i una notificació de fallida és retornada a qui ens ha enviat el missatge a través d'un missatge de FAILURE des de l'Agent AMS. Immediatament que

També es pot esmentar que en el projecte del Pablo García Rué [3] s'ha emprat la tecnologia **SSL** per a obtenir una comunicació segura.

4 Gestió automatitzada d'Alarmes

Actualment, tots els pacients han de realitzar visites periòdiques a l'hospital per a ser examinats per els doctors i controlar la seva evolució. En cada visita, els pacients han d'emplenar un **formulari d'auto avaluació**. En aquest documents, han d'avaluar amb un valor entre 0 i 10, deu aspectes subjectius relacionats amb la salut. Al llarg de la visita el doctor comprova tots aquests valors i els puntua segons el seu criteri d'expert. A més, el doctor també avalua els criteris de complexitat (característiques personals del pacient, problemes mèdics especials, estratègies terapèutiques). Finalment, el doctor pot prendre algunes decisions referents al pacient, com proves clíniques a realitzar-li, modificacions sobre el tractament o canvis d'ubicació (com, per exemple, hospitalitzar a un pacient en un centre soci sanitari si la seva salut s'ha deteriorat massa).

Un dels objectius de *PalliaSys* es facilitar el procés de recollida d'informació de les auto avaluacions dels pacients, de forma que puguin enviar aquesta informació periòdicament des de casa seva (per exemple, cada setmana) sense haver d'anar a l'hospital tan a sovint. La idea és la de posar a la seva disposició les diverses tecnologies de la informació i les comunicacions, com la web, telèfons mòbils o PDAs, de tal forma que cada pacient pugui escollir la forma de comunicació que li sigui més còmode o fàcil d'emprar. Hi ha com uns 500 pacients nous a la *UCP del Hospital de la Santa Creu i de Sant Pau* cada any, això provoca que la quantitat de dades generades per auto avaluacions setmanals sigui molt gran. Per als doctors és una tasca molt costosa estudiar tota aquesta informació i analitzar la evolució de cada pacient en les setmanes o els mesos passats. És per això, que els metges van suggerir que els hi seria molt pràctic el poder tenir una manera automàtica per analitzar tota aquesta informació i ser avisats quan succeeixi una situació anòmala. Més endavant, en la secció de les auto avaluacions, es detalla el procés de recollida d'informació per a les auto avaluacions a través de la pàgina web corresponent.

Per tal facilitar l'anàlisi i l'estudi de les auto avaluacions, i seguint les demandes dels doctors, es va crear un sistema automàtic d'alarmes. Els **Agents Metge** podran generar les alarmes que desitgin i, un cop generades, les assignaran als **Agents Pacient** escollits. Aquests seran els encarregats de mirar si es validen i, en el cas de que així sigui, d'avisar a l'Agent Metge corresponent. Per a entendre millor la definició de les Alarmes, en primer lloc explicarem les auto avaluacions, per a continuar després amb els diferents tipus d'alarmes i el seu cicle de vida.

4.1 Auto Avaluació

Una **auto avaluació** d'un pacient està formada per 10 símptomes que poden portar associada una valoració de 0 a 10. Aquests són: *dolor, debilitat, depressió, ansietat, nàusees, somnolència, gana, benestar, dificultat respiratòria i sequedat de boca*. En cada avaluació, un símptoma només podrà tenir associat un valor que vindrà a representar el grau en el que es manifesta aquest en el pacient. És a dir, si un pacient diu que de Dolor té 2, voldrà dir que en té molt poc. En canvi, si escolleix 8, significarà que en té molt. Cal recordar que el pacient entrarà les avaluacions a través de la interfície

web pertanyent al projecte del Pablo García Rué [3]. A la **Figura 9** es pot veure un exemple d'una possible auto avaluació.

Evaluación

Fecha: 12 / 05 / 2006

Persona que informa: Paciente

PROBLEMA	VAS (0-10)
Dolor	1
Debilidad	9
Depresión	0
Ansiedad	2
Náuseas	10
Somnolencia	4
Apetito	2
Bienestar	3
Dificultad Respiratoria	4
Sequedad de boca	5

Figura 9 - Pantalla d'entrada de les auto avaluacions

A la Base de Dades es guarda un historial amb totes les auto avaluacions que han realitzat els diferents pacients. Això és el que ens possibilita poder treballar amb alarmes d'evolució. Si només poguéssim disposar de l'última auto avaluació, solament podríem tenir alarmes bàsiques, ja que aquestes es centren en la última auto avaluació. Tant les alarmes d'evolució com les bàsiques, s'expliquen en el següent apartat.

4.2 Tipus d'Alarmes

Fins ara, s'havia parlat de que cada **Agent Metge** pot definir les seves pròpies alarmes, assignar-les als **Agents Pacients** que tingui associats i que quan aquests rebin un avís de que un pacient ha realitzat una auto avaluació, les analitzaran i, si es compleixen, faran saltar l'Alarma Corresponent. Però no s'havia dit res de com són aquestes alarmes i com es validen. Això és el que s'explica en aquest apartat.

Les alarmes es poden diferenciar en dues categories, les **Alarmes Bàsiques** (AB) i les **Alarmes d'Evolució** (AE). En ambdós casos, dintre d'una mateixa categoria, aquestes es podran combinar entre elles per tal de formar noves alarmes. Allò que diferencia les Alarmes Bàsiques de les Alarmes d'Evolució és les dades sobre les que treballaran. Les Alarmes Bàsiques analitzaran la informació proporcionada per el pacient en la última avaluació enviada al sistema.

En canvi, una alarma d'evolució comprova el grau de canvi de qualsevol dels deu aspectes descrits en una auto avaluació, considerant les n auto avaluacions passades (*alarmes específiques d'evolució*) o les auto avaluacions que s'han enviat en els d dies passats (*alarmes temporals d'evolució*).

Cal recordar que les alarmes bàsiques i les alarmes d'evolució, no es poden mesclar entre elles.

4.2.1 Alarmes Bàsiques

Les **Alarmes Bàsiques** (AB) són aquelles que tracten la informació proporcionada per el pacient en la última avaluació enviada al sistema. Estan formades per conceptes de les auto avaluacions, amb un *rang de valors*, combinats entre ells per operadors booleans “i”(AND), “o”(OR), “no”(NOT). Una Alarma Bàsica podria tenir el següent format:

$$(\text{Debilitat} > 7) \text{ i } (\text{Dolor} > 8) \Rightarrow \text{Debilitat extrema}$$

El metge podria considerar que en el cas de que la debilitat sigui més gran que 7 i el dolor més gran que 8 és una situació que s'hauria de controlar i llavors podria definir una alarma bàsica que anomenaria *Debilitat extrema* que l'ajudaria a detectar aquest cas tan aviat com es produís. Per a calcular si una alarma d'aquest tipus es compleix, es mirarà que en la última auto avaluació del pacient, aquests símptomes estiguin dintre del rang de valors especificat. En aquest cas, per exemple, es compliria si el pacient hagués especificat que de dolor tenia 9, i de debilitat 8.

Cada alarma definida s'associarà a un **identificador** per tal de que quan aquesta s'activi, el doctor puguis saber immediatament quina és la situació anòmla detectada. Després de definir la situació de Debilitat extrema que s'ha mostrat més amunt, pot definir una nova alarma bàsica combinant aquesta amb noves condicions o amb una altra alarma bàsica, per exemple:

$$(\text{Gana} < 3) \text{ y } \text{Debilitat extrema} \Rightarrow \text{Debilitat perillosa}$$

D'aquesta forma, estem definint una nova situació d'alarma que s'anomenarà *Debilitat Perillosa*. Aquesta s'activarà quan un pacient envii una auto avaluació amb graus molt alts de debilitat i de dolor i un grau molt baix de gana. Aquesta possibilitat de recombinar les alarmes, permet als metges definir una jerarquia completament personalitzada i precisa per a les situacions d'alarma. A més, el fet de que pugui decidir a quins pacients associar cada alarma fa que es pugui personalitzar la supervisió de cada pacient particular. Las alarmes bàsiques permeses son las que se poden generar con la gramàtica següent descrita en BNF:

AlarmaBàsica $\rightarrow$ Condiciones $\Rightarrow$ IdAlarmaBàsica

Condiciones $\rightarrow$ (Cond AND Cond) | (Cond OR Cond) | (NOT Cond) | Cond

Cond $\rightarrow$ Item Comparació Valor | IdAlarmaBàsica

IdAlarmaBàsica $\rightarrow$ Cadena de caràcters

Item $\rightarrow$ dolor | debilitat | depressió | ansietat | nàusees | somnolència | gana | benestar | problemes de respiració | sequedat de boca

Comparació → < | >

Valor → 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

A la **Figura 10** podem veure un exemple d'estructuració d'una alarma seguint aquesta gramàtica.

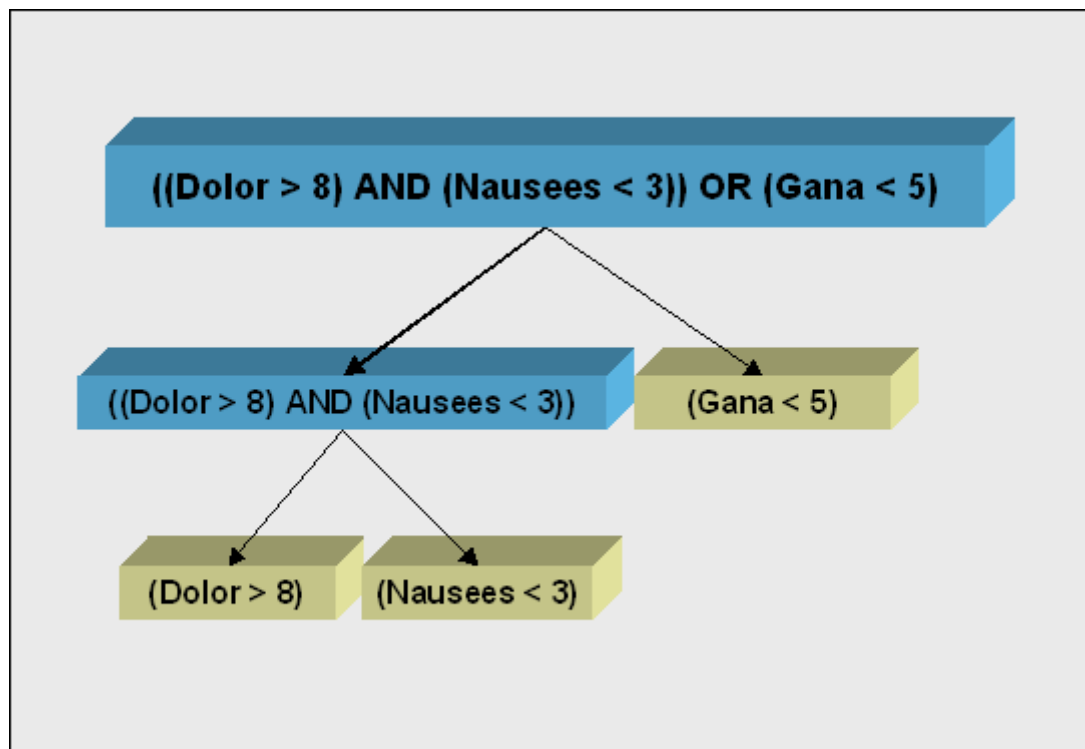


Figura 10 - Exemple d'estructuració d'una Alarma

4.2.2 Alarmes d'Evolució

Les **Alarmes d'Evolució** (AE) es van introduir ja que en molts casos pot ser més útil o interessant estudiar l'evolució del estat de salut d'un pacient al llarg d'una sèrie d'auto avaluacions o dies que no senzillament analitzar les dades que ens proporciona l'última auto avaluació. Per a posar un exemple, ens podríem trobar en el cas de que un pacient especificqués un grau 6 per al dolor, cosa que ens podria no semblar alarmant a no ser que ens poguéssim trobar en el cas de que la setmana abans o dues auto avaluacions abans, el seu grau de dolor hagués estat molt més baix, per exemple 2. En aquest cas, ho podríem considerar anormal i preocupant.

La funció d'una Alarma d'Evolució és la de comprovar el grau de canvi en qualsevol dels deu aspectes que es contemplen en una auto avaluació, considerant les n auto avaluacions passades (**alarmes específiques d'evolució**) o les auto avaluacions que s'han enviat en el d dies passats (**alarmes temporals d'evolució**). Un exemple d'una alarma específica d'evolució pot ser el següent:

(2 avals.) Δ Debilitat $> 3 \Rightarrow$ Augment ràpid de la debilitat

En aquest cas, l'alarma *Augment ràpid de la debilitat* s'activaria quan ens trobéssim que el pacient, en la última auto avaluació, informa d'un grau de debilitat de com a mínim 4 unitats més gran que la que està anotada en la auto avaluació passada. Això es donaria, per exemple, si en la auto avaluació anterior el pacient hagués especificat que de debilitat tenia 2 i en l'actual especificués que el seu grau de debilitat era 6. Per les alarmes temporals d'evolució el que es fa es comprovar l'evolució del pacient al llarg d'un període de temps. Un exemple d'una alarma d'aquest tipus podria ser el següent:

(28 dies) Δ Dolor $> 4 \Rightarrow$ Augment extrem del dolor

L'alarma "Augment Extrem del dolor" s'activaria quan, després de rebre una auto avaluació d'un pacient i analitzar totes les auto avaluacions enviades per aquest en els 28 dies anteriors, es veïés que el dolor ha patit un increment de 4 punts en la última auto avaluació respecte la primera auto avaluació. Aquesta evolució ha de ser continua, és a dir, d'una avaluació a una altra sempre s'ha d'incrementar o, com a molt ha de mantenir el valor, però mai decrementar-se. Per exemple, si en aquest període tinguéssim 5 auto avaluacions, un cas en el que s'activaria l'alarma seria que els graus de dolor haguessin estat 2,3,5,5,6. Si el que es mirés en l'alarma fos el decrement d'algun símptoma la filosofia seria la mateixa, que d'una avaluació a una altra sempre es decrementés o bé es mantingués el valor, però que mai s'incrementés. Les alarmes de evolució permeses son les que es poden generar amb la següent gramàtica, descrita en BNF:

AlarmaEvolució $\rightarrow$ (enter "avals.") Condiciones $\Rightarrow$ IdAlarmaEspecificaEvolució |
(enter "dies") Condiciones $\Rightarrow$ IdAlarmaTemporalEvolució

Condicions $\rightarrow$ (Cond AND Cond) | (Cond OR Cond) | (NOT Cond) | Cond

Cond $\rightarrow$ Δ Item Comparació Valor | IdAlarmaEspecificaEvolució |
IdAlarmaTemporalEvolució

IdAlarmaEspecificaEvolució $\rightarrow$ Cadena de caràcters

IdAlarmaTemporalEvolució $\rightarrow$ Cadena de caràcters

Item $\rightarrow$ dolor | debilitat | depressió | ansietat | nàusees | somnolència | gana | benestar |
problemes de respiració | sequedat de boca

Comparació $\rightarrow < | >$

Valor $\rightarrow 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

A l'igual que amb les alarmes bàsiques, si observem la descripció que acabem de formula podem veure que els metges també poden combinar l'anàlisi de diversos aspectes en una mateixa alarma d'evolució i combinar les alarmes entre elles. El criteri d'assignació als pacients és també el mateix que per les alarmes bàsiques.

4.3 Alarmes: Gestió i Cicle de Vida

El cicle de vida d'una alarma, sigui Bàsica o d'Evolució es divideix en les quatre fases següents: *Definició*, *Assignació*, *Validació* i *Activació*.

4.3.1 Definició

Cada Metge associat a la UPC, a través de el seu **Agent Metge** té la possibilitat de crear les seves pròpies alarmes, per a fer-ho disposarà de dues interfícies, una per a les *Alarmes Bàsiques* (**Figura 11**) i una altra per a les *Alarmes d'Evolució* (**Figura 12**).

Edición Alarmas Básicas

Nombre de la alarma: Debilitat Extrema

Condición

Problema: Dolor

0 1 2 3 4 5 6 7 8 9 10

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☒ ☒

Agregar Limpiar Seleccionar todos

Debilidad:8,9,10
Dolor:9,10
Debilitat Extrema

AND
OR
NOT
Eliminar
Importar alarma...

Condición seleccionada
(Debilidad:8,9,10) AND (Dolor:9,10)

Crear Cerrar

Figura 11 - Pantalla edició Alarmes Bàsiques

Aquestes interfícies proveeixen als metges un sistema amb la capacitat de combinar les alarmes existents per a definir alarmes més complexes. Un cop hagi definit una alarma, aquesta quedarà guardada i preparada per a poder-se assignar a un pacient. L'**Agent Analitzador** permetrà la creació d'alarmes que seran visibles per tots els Agents Metges i que podran fer servir com a seves.

A l'hora de definir les alarmes, per tal de facilitar la tasca als metges i fer-ho més intuïtiu, s'ha optat per un lleuger canvi respecte la notació emprada en els apartats

anteriors. A pesar d'això, les alarmes continuen sent les mateixes. Per posar un exemple, la alarma bàsica que havíem comentat abans com a *Debilitat extrema*:

(Debilitat > 7) i (Dolor > 8)

Ara passaria a ser:

(Debilitat: 8,9,10) AND (Dolor: 9,10)

Edición Alarmas de Evolución

Nombre de la alarma: Augment ràpid de la debilitat

Condición

Problema	Variación	Período	Unidad
Sequedad de boca	<	1	Días

Agregar

Augment ràpid de la debilitat
Variación Sequedad de boca < 1 por 2 Días

AND
OR
NOT
Eliminar
Importar alarma...

Condición seleccionada
Variación Debilidad > 3 por 2 Evaluaciones

Crear Cerrar

Figura 12 – Pantalla edició Alarmes Evolució

En el cas d'una Alarma d'Increment, com la d'*Augment ràpid de la debilitat*, ens trobaríem que abans era:

(2 avals.) Δ Debilitat > 3 => Augment ràpid de la debilitat

I ara passaria a ser:

Variació Debilitat > 3 per 2 Avaluacions

4.3.2 Assignació

Un metge podrà anar generant les alarmes que desitgi sense la necessitat d'assignar-les immediatament a cap **Agent Pacient**. En qualsevol moment, aquesta alarmes podran ser assignades a qualsevol Agent Pacient que tingui associat l'**Agent Metge**. Cada alarma la es podrà assignar a tants Agents Pacients com es vulgui i un Agent Pacient pot tenir assignada més d'una alarma. Per a realitzar el procés d'assignació l'Agent Metge disposa de la pantalla que podem veure a la **Figura 13**. Per al procés d'assignació farem servir el protocol FIPA Subscribe que s'ha explicat en l'apartat d'Arquitectura del sistema.

Asignación Alarmas

Agentes Paciente activos

Id Agente	Nombre paciente
11111	Jordi Sabaté

Refrescar Crear Agente Paciente

Detalles Agente Paciente

Id. Agente: 11111 Mostrar

Debilitat Extrema
Augment ràpid de la debilitat

{Debilitad:8,9,10} AND {Dolor:9,10}

Agregar Alarma Eliminar Alarma

Figura 13 - Pantalla assignació Alarmes

4.3.3 Validació

Quan un pacient realitza una auto avaluació, aquesta s'emmagatzema en la Base de Dades de la UCP i l'**Agent Pacient** rep l'avís per part de l'**Agent Socket** de que el seu pacient ha realitzat una auto avaluació, arribat aquest moment, verificarà que no es compleixin les alarmes que posseeix. Per a fer-ho, primer sol·licitarà les dades de la avaluació a l'**Agent DBWrapper** i comprovarà si totes les Alarmes Bàsiques que té emmagatzemades es compleixen o no. En el cas de que també hi hagin Alarmes d'Evolució també haurà de sol·licitar les dades de les auto avaluacions anteriors necessàries per a poder comprovar l'evolució del pacient. Per raons de seguretat, els Agents Pacient no guarden les auto avaluacions, aquestes s'emmagatzemen únicament en la Base de Dades de la UCP.

4.3.4 Activació

En el cas de que alguna de les alarmes es validi, aquesta s'activarà i s'enviarà una notificació al Agent Metge corresponent amb l'Identificador d'aquesta i la del pacient que l'ha fet activar. També es guardarà en la Base de Dades com a Alarma activada a la taula d'**Alarmes Activades**. Quan l'Agent Doctor rebí una notificació de que s'ha activat una Alarma, avisarà al Metge mitjançant una finestra emergent modal de que s'ha produït una alarma. El Metge podrà portar un control de les alarmes activades a través d'una interfície que proporcionarà un llistat de les alarmes amb les seves característiques, estudiar-les i decidir quina és la millor conducta a seguir (per exemple, trucar al pacient, enviar un equip a examinar-lo, o enviar un missatge al pacient sol·licitant una visita en la UCP de l'hospital com abans millor). Un cop hagi decidit quina actuació ha de realitzar, pot decidir passar l'alarma a històrica.

5 Instal·lació / Manual de l'Administrador

En aquest apartat entrarem en detall en com obtenir, instal·lar i configurar els diferents elements que es requereixen per a poder fer funcionar correctament el projecte. En primer lloc passarem a descriure com instal·lar el projecte del Pablo García Rué [3] ja que aquest el Sistema Multi Agent depèn d'aquest per a poder funcionar correctament. Un cop es tingui instal·lada la part del Pablo, es procedirà a explicar com instal·lar el Sistema Multi Agent. En últim lloc també s'explicarà una mica com realitzar la posta en marxa.

5.1 Requeriments del Sistema

Les instruccions d'instal·lació, així com la tecnologia emprada fa que sigui necessari l'utilització de Windows com a sistema operatiu, i un mínim d'un Pentium 4 o equivalent amb 512 Mb de RAM. Aquestes especificacions es refereixen a la màquina en que es va provar, és possible que amb configuracions inferiors també funcioni, però no es garanteix un funcionament eficient.

5.2 Instal·lació del Projecte del Pablo

Per a poder fer funcionar tot el sistema en primer lloc seria necessari que a la màquina hi hagués instal·lat el projecte del *Pablo García Rué* [3]. Per a aquest propòsit necessitarem disposar del **Servidor Web Apache** amb el mòdul per a **PHP** i el mòdul de seguretat **SSL**, també la **Base de Dades** i la **connexió ODBC** i, finalment, la aplicació final dintre del servidor Web.

S'ha de tenir en compte que les versions del programari poden variar i que en aquest document s'especificaran les que es van fer servir per a provar la aplicació en el seu moment.

5.2.1 El Servidor Web Apache

De l'**Apache** es disposen de diverses versions, la que es va fer servir per a aquest projecte i que no va generar cap problema va ser la 1.3.x per a Windows. Per a aconseguir la última versió ens podem dirigir a la pàgina web d'Apache:

<http://httpd.apache.org>

També podem trobar documentació de les diferents versions d'Apache. La més recent la podem trobar a la pàgina:

<http://httpd.apache.org/docs-project/>

Obtindrem la versió d'Apache que necessitem per al nostre sistema operatiu des de la pàgina:

<http://httpd.apache.org/download.cgi>

En el nostre cas vam fer servir la versió *apache_1.3.27-win32-x86-no_src.msi*. Com ja s'ha comentat anteriorment, aquesta versió serveix com a referència per a saber el fitxer que s'ha d'escollir, però haurem d'agafar la versió més moderna de l'Apache versió 1.3.x, que segurament no es correspondrà amb aquesta.

A l'instal·lar-lo se'ns demanaran les dades del nostre servidor: el *Network Domain*, el *Server Name* i l'*Administrator's Email Address*. Per exemple:

localhost/localhost/marta.marin@estudiants.urv.cat

Escollim instal·lar-lo com un servei (*Run as a service for all users - recommended*).

Després escollim la instal·lació Completa i l'instal·lem en el directori **C:\Apache**.

Ara ja tenim l'Apache Instal·lat. En la configuració per defecte el Servidor Web es posa en marxa per defecte a l'engegar el Windows. Per a comprovar que el tenim instal·lat correctament, obrim l'explorador i posem `http://localhost`. Si està funcionant se'ns mostrarà una pàgina de benvinguda.

5.2.2 Mòdul SSL sobre Apache

La versió que ve a continuació és per a la versió 1.3.x, la que acabem d'instal·lar. I per a instal·lar-lo necessitem haver instal·lat prèviament l'Apache.

5.2.2.1 Primera configuració sobre Apache

Editem el fitxer **httpd.conf** que es troba al directori *C:\Apache\Apache\conf*. Comentem la línia on posa "*port 80*" posant-li # davant. Escrivim *Listen 80* i *Listen 443*.

En el *ServerName* posem el nom del nostre servidor. Reiniciem el servidor i comprovem que de moment funciona fent `http://localhost:443`.

5.2.2.2 Instal·lació del mod_ssl i OpenSSL

Necessitem dels paquets **mod_ssl** i **OpenSSL** per a dotar de seguretat al nostre servidor. Ens els podem baixar de: <http://www.modssl.org/contrib/>

En el nostre cas hem fet servir el fitxer *Apache_1.3.27-Mod_SSL_2.8.14-Openssl_0.9.7b-Win32.zip*. El descomprimim i cerquem els fitxers **ssleay32.dll** i **libeay32.dll** i els col·loquem al directori de Windows amb les altres dll. Això variarà depenent de la versió de Windows.

Finalment descarreguem el fitxer *http://tud.at/programm/openssl.cnf* i el copiem on haguem descomprimit el *Openssl-0.9.7a-Win32*.

5.2.2.3 Creació d'un certificat de test

Des d'una finestra de comandes ens dirigim on haguem descomprimit el *Openssl-0.9.7a-Win32* i executem:

```
>> openssl req -config openssl.cnf -new -out my-server.csr
```

Això ens generarà un *certificate signing request* i una clau privada **RSA** de 1024 bits en el fitxer **privkey.pem**. Al llarg del procés ens anirà preguntant una sèrie de dades que haurem d'omplenar. Sobretot s'ha de tenir en compte que el nom del servidor coincideixi amb el real. Amb tot això haurem creat el fitxer **my-server.csr** i el **privkey.pem** (aquest també en **.rnd**).

```
>> openssl rsa -in privkey.pem -out my-server.key
```

Ara ja tenim el fitxer **my-server.key**, solament llegible per l'Apache i l'administrador. Esborrem el fitxer **.rnd**

```
>> openssl x509 -in my-server.csr -out my-server.cert -req -signkey my-server.key -days 3650
```

Amb això tindrem el fitxer **my-server.cert**. Finalment creem el directori *c:/Apache/Apache/conf/ssl* i movem **my-server.key** i **my-server.cert** a aquest directori acabat de crear.

5.2.2.4 Configuració d'Apache i mod_ssl

Amb l'Apache parat, copiem els fitxers executables (.exe, .dll, .so) que trobem dins del directori *Apache_1.3.27-Mod_SSL_2.8.14-Openssl_0.9.7b-Win32* a directori arrel de l'Apache (*c:/Apache/Apache*).

Copiem el fitxer *mod_ssl.so* des de *Apache_1.3.27-Mod_SSL_2.8.14-Openssl_0.9.7b-Win32/modules* a *c:/Apache/Apache*.

Al **httpd.conf** afegim:

```
(després dels LoadModules)
LoadModule ssl_module modules/mod_ssl.so
(després dels AddModule)
AddModule mod_ssl.c
(Al final del fitxer)
SSLMutex sem
SSLRandomSeed startup builtin
SSLSessionCache none
SSLLog logs/SSL.log
```

SSLLogLevel info

```
<VirtualHost dominiServidor:443>  
SSLEngine On  
SSLCertificateFile conf/ssl/my-server.cert  
SSLCertificateKeyFile conf/ssl/my-server.key  
</VirtualHost>
```

Per a engegar l'**Apache** farem servir la següent configuració: `C:\Apache\Apache\Apache.exe -w -n "Apache" -k start -D SSL`. Posem en marxa l'**Apache** i fent servir <https://localhost> accedim al servidor segur.

5.2.3 PHP sobre Apache

Podem trobar la versió que necessitem a la pàgina <http://www.php.net>. En el nostre cas hem fet servir la versió *php-4.2.3-Win32.zip*.

El descomprimim en `C:\PHP\` obtenint `C:\PHP\ php-4.2.3-Win32` i seguirem les instruccions que trobarem en el fitxer **install.txt**.

5.2.3.1 Modificacions sobre PHP

Dintre del fitxer `C:\windows\php.ini`:

```
Register_globals = Off  
Magic_quotes_sybase = On
```

Per a les sessions amb **PHP** haurem de crear-nos el directori `/tmp` a l'arrel. I en el **php.ini** posar:

```
session.save_path = c:/tmp  
sessiion.use_cookies = 1
```

5.2.4 Base de Dades i Connexió ODBC

Aquestes instruccions canvien depenent de la versió del Windows que es faci servir i serveixen a mode d'exemple. Per a crear la connexió ODBC:

- *Mi Pc -> Panel de Control -> ODBC*
- Anem a la pestanya **DNS de Sistema** i escollim *Agregar*.
- Seleccionem el **Microsoft Access Driver**
- En el nom del *Origen de Dades*, posem el nom amb el que anomenarem a la Base de Dades: '**ucp**'
- En la *Base de Dades*, indiquem on està ubicat el fitxer *.mdb*
- Acceptem tot i ja tenim creada la connexió ODBC

5.2.5 Instal·lació de l'Aplicació Web

És relativament senzill, només necessitem col·locar tots els fitxers dintre del directori htdocs d'Apache. Així podrem accedir a ella de la següent forma:
<https://servidor/index.htm>

5.3 Instal·lació del Sistema Multi Agent.

Quan ja tinguem instal·lat el projecte del Pablo García Rué [3] els fitxers i la BD de dades adequada, procedirem a instal·lar el Sistema Multi Agent, és a dir el projecte al que fa referència aquesta documentació.

Primer de tot, necessitarem que el sistema disposi d'un JRE instal·lat, com a mínim la versió 1.4. En cas de no disposar d'ell, aquest es pot aconseguir a la pàgina web de Sun:

<http://www.sun.com/>

Un cop ja tenim el JRE, el que ens queda és copiar els següents fitxers al lloc des d'on volem realitzar l'execució:

- *palliaSys.jar*: S'haurà de copiar a totes les màquines on vulguem executar part del Sistema Multi Agent, és a dir, tant al Servidor (la màquina en la que haguem instal·lat la Base de Dades i l'Aplicatiu Web), com als ordinadors dels metges com en el que vulguem instal·lar l'Agent Analitzador. Aquest fitxer és el que conté les classes de l'aplicació així com totes les llibreries necessàries per a l'execució, com és el cas de les del Jade.
- *plataforma.bat* i *plataforma.conf*: Aquests son els fitxers que contene les comandes per a posar en marxa la plataforma on hi tindrem l'Agent DBWrapper, l'Agent Socket i els Agents Pacient. En conseqüència, només es copiaran al Servidor i al mateix lloc on haguem copiat el fitxer *palliaSys.jar*.
- *agentMetge.bat* i *agentMetge.conf*: S'haurà de copiar als ordinadors personals dels metges que vulguin tenir el seu propi Agent Metge. Com en el cas anterior, també els haurem de copiar on estigui el fitxer *palliaSys.jar*.
- *agentEstadistic.bat* i *agentEstadistic.conf*: Amb aquests fitxers s'engegarà l'Agent Estadístic, en conseqüència, els haurem de copiar a l'ordinador on es vulgui fer funcionar aquest agent. També els copiarem al mateix directori que el *palliaSys.jar*.

Un cop realitzats tots aquests passos, ja tenim el sistema preparat per a iniciar l'execució del **Sistema Multi Agent**.

5.4 Posta en marxa / Execució

5.4.1 Servidor (Agent DBWrapper, Agent Socket i Agents Pacient)

Primer de tot s'ha d'engegar la plataforma on estan l'Agent DBWrapper i l'Agent Socket i on es crearan els diferents Agents Pacient. Aquesta s'haurà d'engegar al Servidor que és on tindrem la Base de Dades que haurem instal·lat prèviament. Per això disposem del fitxer *plataforma.bat* que hem copiat durant el procés d'instal·lació:

```
java -cp palliaSys.jar jade.Boot -mtp  
jade.mtp.http.MessageTransportProtocol(http://adreçaServidor:7788/acc)  
-port portPlataformaServidor  
dbwrapper:paliatius.agents.AgentDBWrapper socket:paliatius.agents.AgentSocket  
-conf plataforma.conf
```

Abans d'executar-lo, però haurem de realitzar un parell de modificacions per tal de configurar-lo i adaptar-lo a l'entorn:

- **adreçaServidor** -> Haurem de canviar-ho per l'adreça real del nostre servidor
- **portPlataformaServidor** -> Especificarem el port on es rebran les connexions dels Agents que tindrem en altres màquines.

5.4.2 Agent Metge

Per a engegar un Agent Metge en l'ordinador personal d'un metge disposem del fitxer *agentMetge.bat*:

```
java -cp palliaSys.jar jade.Boot -mtp  
jade.mtp.http.MessageTransportProtocol(http://picarol:7778/acc)  
idDotactor:paliatius.agents.AgentDoctor (adreçaServidor portPlataformaServidor)  
-conf agentMetge.conf
```

En aquest cas, també ens tocarà modificar l'**adreçaServidor** i el **portPlataformaServidor**. Haurem de posar els valors que haguem especificat en el Servidor. També s'haurà de canviar l'**idDoctor** per el identificador del Metge.

5.4.3 Agent Estadístic

I per a l'Agent Estadístic disposem del fitxer: *agentEstadistic.bat*:

```
java -cp palliaSys.jar jade.Boot -mtp  
jade.mtp.http.MessageTransportProtocol(http://picarol:7778/acc)  
analitzador:paliatius.agents.AgentDataAnalyser (adreçaServidor portPlataformaServidor)  
-conf agentEstadistic.conf
```

A l'igual que amb el cas de l'Agent Metge, també haurem de modificar els paràmetres **adreçaServidor** i el **portPlataformaServidor** amb els especificats al Servidor.

6 Interfície Gràfica / Manual d'Usuari

En aquest apartat es descriurà la interfície gràfica del projecte i el seu funcionament detalladament per a que serveixi de referència com a *Manual d'Usuari*. En el següent apartat també es disposarà d'un *Joc de Proves* que permetrà veure amb més claredat el funcionament i diferents casos d'utilització.

En el projecte es disposa de dos agents amb interfície gràfica que permeten la interacció de l'usuari, en aquest cas Metges i el Cap de la Unitat, amb el sistema. Aquests són l'**Agent Metge** i l'**Agent Analitzador**.

6.1 Agent Metge

Com s'ha comentat en l'apartat d'instal·lació i execució, per a engegar l'Agent Metge farem servir el fitxer *agentEstadistic.bat*.

Un cop engegat l'Agent Metge, el primer que ens apareixerà serà una finestra modal en la que se'ns demanarà que ens validem com a metge. Podem veure la finestra de validació a la **Figura 14**.



Figura 14 - Finestra de Validació

Si la validació és incorrecta, saltarà un missatge d'error **Figura 14**. Un cop validat l'Accés, s'engegarà la interfície de l'Agent Metge. Podem veure-la a la **Figura 16**.

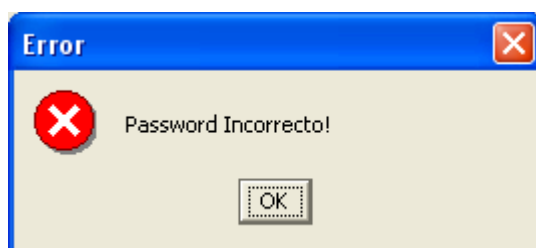


Figura 15 - Error de Validació

Degut a que el procés lògic d'assignació d'alarmes requereix que els Agents Pacient estiguin donats d'alta, començarem explicant la gestió d'aquests, és a dir l'Alta i la Baixa dels Agent Pacient, per a acabar amb la creació i generació de les alarmes, tant de les Bàsiques com de les d'Evolució.

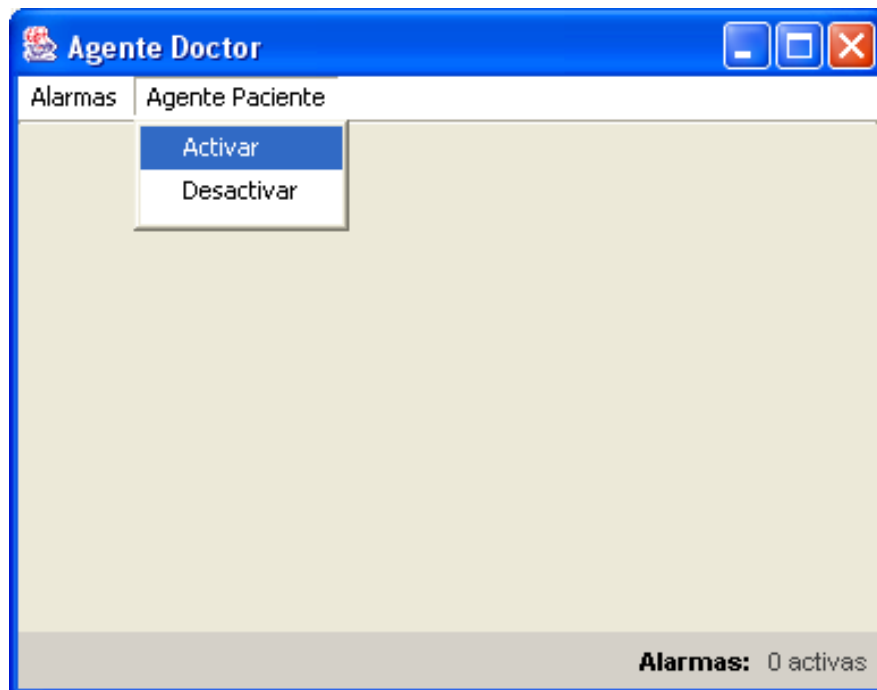


Figura 16 - Agent Metge

En aquesta pantalla podem diferenciar dues zones, la del menú on podem accedir a les opcions de les *Alarmes*:

- **Editar Alarma Bàsica:** Ens obrirà la finestra de disseny de les Alarmes Bàsiques que ens permetrà generar noves Alarmes Bàsiques o recombinar les que ja tinguem.
- **Editar Alarma de Evolució:** Ens obrirà la finestra corresponent al disseny de les Alarmes d'Evolució. A l'igual que l'anterior, podrem crear-ne de noves o recombinar les que hàgim definit prèviament.
- **Asignar:** Obre la pantalla que ens permetrà realitzar l'assignació d'alarmes als Agents Pacient que tinguem associats. L'Agent Pacient l'haurem d'haver donat d'alta prèviament.
- **Alarmas Activas:** Porta a un llistat de les Alarmes que estan actives en aquest moment. Des d'aquest llistat podrem passar les Alarmes a Històriques.
- **Historial de Alarmas:** Des d'aquí s'accedeix a l'historial de les alarmes dels dos últims mesos. Apareixeran tant les actives com les que ja han estat marcades com a històriques. Des de la mateixa finestra es podran recuperar alarmes d'altres dates.

I també a les opcions de *Gestió dels Agents Pacient*:

- **Activar:** Obrirà la finestra modal que ens permetrà donar d'alta un *Agent Pacient*. Cal recordar que un *Agent Pacient* estarà associat a un sol pacient i a l'invers i que un *Agent Pacient* només pot estar assignat a un sol *Agent Metge*.
- **Desactivar:** També ens obrirà una altra finestra modal, aquesta, però, tindrà la funció de desactivar a un *Agent Pacient* que s'hagi donat d'alta prèviament.

En la part inferior de la pantalla disposarem d'una barra on es mostrarà informació referent a les alarmes que estiguin actives en aquell moment. Clicant a sobre se'ns obrirà el mateix llistat d'Alarmes Actives al que podem accedir des del menú d'Alarmes. En aquesta barra se'ns mostrarà en tot moment i actualitzat, el nombre d'**Alarmes Actives** que tenim pendents de passar a **Històriques**.

Ara anem a veure com es realitza la Gestió d'Agents Pacient, tant l'Alta com la Baixa, per continuar després amb la gestió de les Alarmes.

6.1.1 Gestió dels Agents Pacient

Tal i com s'ha comentat amb anterioritat, els Agents Metges són els encarregats de gestionar la creació i eliminació dels Agents Pacient. Cal recordar que un *Agent Pacient* només podrà estar associat a un *Agent Metge* i que serà necessari que el pacient al que faci referència s'hagi donat d'alta prèviament al sistema. Per cada pacient només hi haurà un *Agent Pacient*.

6.1.1.1 Alta Agent Pacient

Per a donar d'alta un *Agent Pacient* hi ha dos punts d'accés, un des del menú: "*Agente Paciente -> Activar*" i l'altre des de la pantalla d'assignació d'alarmes a la que s'hi arriba des de: "*Alarmas->Asignar*". En la **Figura 17** podem veure la pantalla d'Alta d'un *Agent Pacient*.

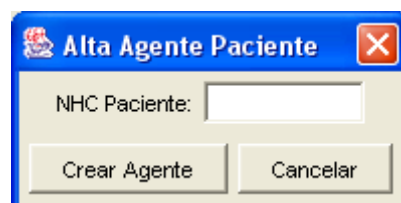


Figura 17 - Pantalla Alta Agent Pacient

Prèviament a realitzar l'alta, es porten a terme una sèrie de comprovacions. En primer lloc es comprova que existeixi el NHC, és a dir, que tinguem un pacient amb l'NHC donat. Un *Agent Pacient* no pot existir a no ser que amb anterioritat s'hagi donat d'alta el pacient al sistema. En segon lloc es comprova que l'*Agent Pacient* no existeixi. En el cas de que ja hagi estat creat, també es mira que no pertanyi a un altre *Metge*. Si en el procés de realitzar aquestes validacions es comprova que una no es compleix, llavors es generarà un missatge d'alerta com el de la **Figura 18** i es procedirà a cancel·lar l'operació d'Alta.

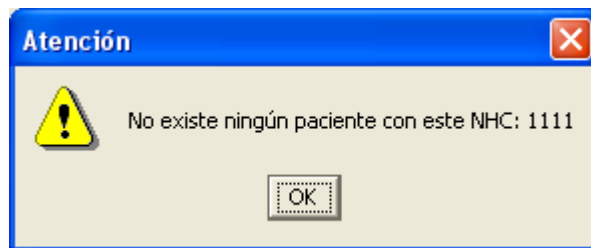


Figura 18 - Missatge Error en la Creació de l'Agent Pacient

6.1.1.2 Baixa Agent Pacient

Per a donar de baixa un Agent Pacient ho podem fer des del menú: *Agente Paciente -> Desactivar*. Ens apareixerà una finestra emergent molt similar a la de l'Alta. En ella hem d'introduir el NIF del pacient al que tenim associat l'Agent i clicar sobre "Eliminar Agent".

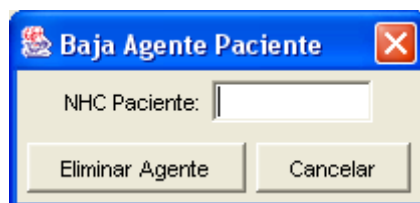


Figura 19 - Pantalla Baixa Agent Pacient

Donar de Baixa a un Agent Pacient implica que s'eliminaran totes les alarmes que se li havien associat i que qualsevol altre Agent Metge podrà donar-lo d'alta lliurement.

Durant el procés de Baixa, es porten a terme una sèrie de comprovacions com que l'Agent estigui actiu. En el cas de que estigui actiu, també es comprovarà que pertanyi a l'Agent Metge que l'està intentant donar de Baixa. En el cas que no estigués actiu o bé fos d'un altre Agent Metge, saltaria un missatge d'alerta avisant del que està succeint.

6.1.2 Generació d'Alarmes

L'Agent Metge posseeix la capacitat de generar alarmes i assignar-les als Agents Pacient que tingui associats. Degut a que es disposa de dos tipus d'Alarmes: les Bàsiques i les d'Evolució, es tindran dues pantalles per a la generació d'alarmes, una per a les Alarmes Bàsiques i una altra per a les Alarmes d'Evolució. En aquest apartat començarem explicant com es crea una Alarma Bàsica, primer simple i després com es pot combinar per obtenir-ne de complexes. Després farem el mateix per a les alarmes d'Evolució.

6.1.2.1 Generació d'una Alarma Bàsica

Una Alarma Bàsica, tal i com hem comentat anteriorment, és la que analitza la informació de la última auto avaluació realitzada pel pacient i comprova si algun dels valors que ha assignat el pacient als problemes coincideix amb el rang de valors que validarien l'Alarma. Una Alarma Bàsica està formada per la combinació de diverses condicions (Problema més valors) amb OR, NOT i AND. Aquestes Alarmes, un cop generades es poden recombinar per crear-ne de noves. Primer veurem com generar una Alarma Bàsica Simple i també com fer-ho per una Alarma Bàsica Complexa.

A la **Figura 20** es pot veure la pantalla de generació de les Alarmes Bàsiques. Per arribar a aquesta pantalla ho farem mitjançant el menú: *Alarmas -> Editar Alarma Bàsica*

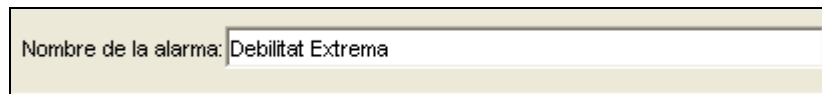
The screenshot shows a software window titled "Edición Alarmas Básicas". It contains the following elements:

- Nombre de la alarma:** A text field containing "Debilitat Extrema".
- Condición:** A section with a "Problema" dropdown menu set to "Dolor". To its right is a row of checkboxes labeled 0 through 10. Checkboxes for 9 and 10 are checked.
- Buttons:** "Agregar", "Limpiar", and "Seleccionar todos" are below the checkboxes. To the right of the list are buttons for "AND", "OR", "NOT", "Eliminar", and "Importar alarma...".
- List of conditions:** A list box on the left contains "Debilidad:8,9,10", "Dolor:9,10", and "Debilitat Extrema" (which is highlighted).
- Condición seleccionada:** A text area at the bottom shows the logical expression "(Debilidad:8,9,10) AND (Dolor:9,10)".
- Bottom buttons:** "Crear" and "Cerrar".

Figura 20 - Pantalla de Generació d'Alarmes Condicionals

Podem dividir la pantalla en quatre zones en les que es repartiran els passos que haurem de realitzar per tal de generar l'Alarma Bàsica:

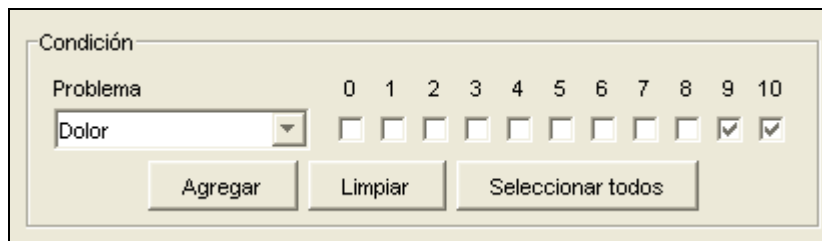
- 1) *Nom de l'Alarma*: Aquí especificarem el nom que li vulguem donar a l'Alarma Bàsica, aquest nom serà el que farem servir també com a identificador de la mateixa. Podem apreciar el detall en la **Figura 21**.



Nombre de la alarma: Debilitat Extrema

Figura 21 - Detall Pantalla Alarmes Bàsiques: Nom Alarma

- 2) *Condició*: En segon terme es té la secció on es poden definir Condicions Simples, problema més valors (**Figura 22**). En el combo del problema s'escull el símptoma sobre el que mirarem la condició. Les caselles de selecció ens permeten seleccionar els valors del símptoma que activaran la Condició, aquests van del 0 al 10. També disposem de tres botons, el de *Netejar* deixarà en blanc les caselles, el *Seleccionar tots* les marcarà totes i l'*Agregar* afegirà la condició a la *Pissarra de Composició*.



Condición

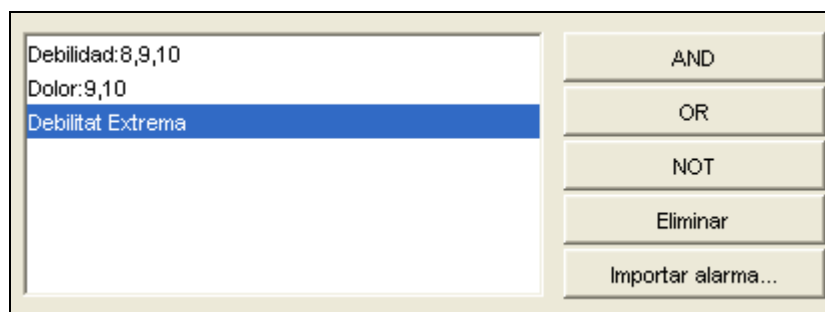
Problema	0	1	2	3	4	5	6	7	8	9	10
Dolor	☐	☐	☐	☐	☐	☐	☐	☐	☐	☒	☒

Agregar Limpiar Seleccionar todos

Figura 22 - Detall Pantalla Alarmes Bàsiques: Definició Condicions

Aquí és on hem d'anar generant les condicions que vulguem incloure a la nostra alarma simple i anar-les agregant a la pissarra per a poder-les combinar més endavant.

- 3) *Pissarra de Combinació*: En aquest espai podrem combinar les condicions entre elles mitjançant els botons *AND*, *OR* i *NOT*. Per a combinar dues regles amb un *AND* o amb un *OR* haurem de seleccionar un mínim de dues regles, per a seleccionar-les ho farem pressionant la tecla *Shift* i clicant sobre les condicions escollides. Per al *NOT* només hem de seleccionar una sola condició.



Debilitad:8,9,10	AND OR NOT Eliminar Importar alarma...
Dolor:9,10	
Debilitat Extrema	

Figura 23 - Detall Pantalla Alarmes Bàsiques: Zona Pissarra

En el cas que a l'intentar combinar les condicions no tinguem seleccionat el nombre correcte, és a dir, en el cas de que vulguem, per exemple,

aplicar un AND i només tinguem seleccionada una condició, ens sortirà una pantalla d'avís com la que es pot veure a la **Figura 24**. El botó *Eliminar* ens permet eliminar una condició que no ens interessi de la pissarra.

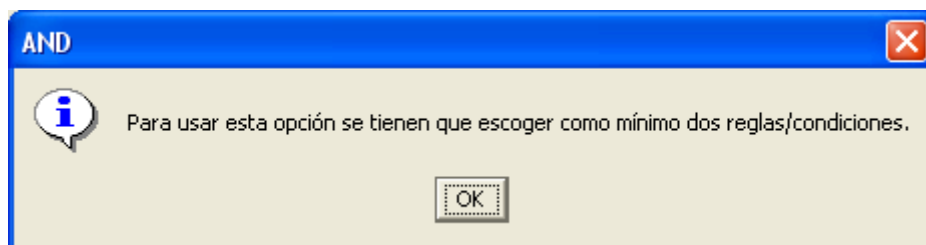


Figura 24 - Error a l'hora de combinar condicions

Amb *Importar alarma...* se'ns obrirà una pantalla modal que ens permetrà importar alarmes definides prèviament a la zona de la pissarra. En la **Figura 25** podem veure la finestra que ens sortirà per a importar les alarmes.

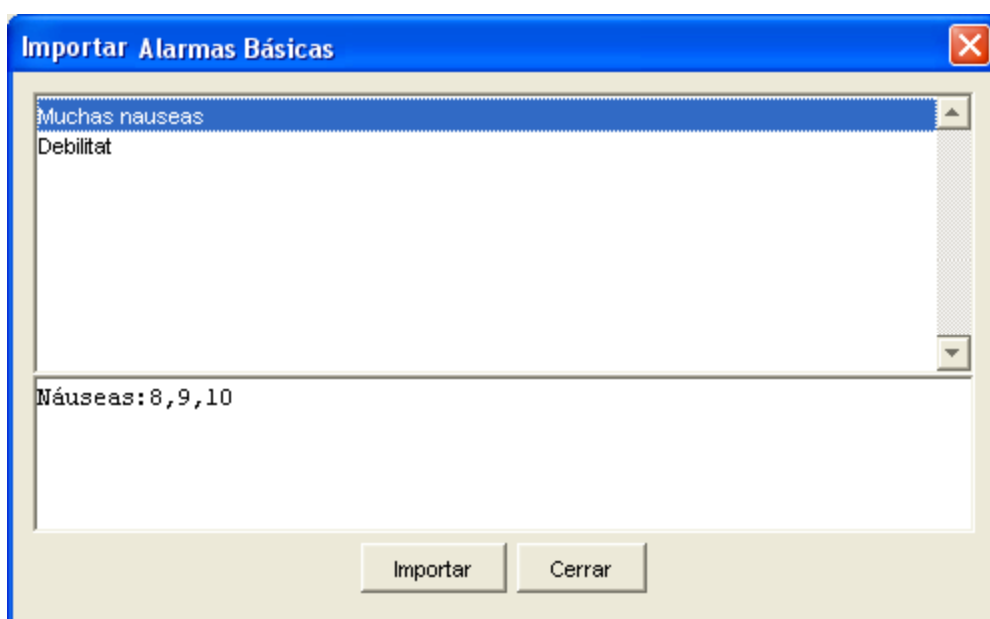


Figura 25 - Pantalla Importar Alarmes Bàsiques

Aquesta pantalla estarà dividida en dues parts, en la superior ens apareixerà el llistat de les alarmes disponibles i en la inferior l'equació de l'alarma que tinguem seleccionada.

- 4) *Condició Seleccionada*: En aquesta part (**Figura 25**) ens apareixerà sempre la condició o l'alarma que tinguem seleccionada en la pissarra de condicions. Aquesta serà l'equació de l'Alarma que es guardarà quan pitgem al botó *Crear*. En el cas de que el que hi hagi seleccionat sigui una alarma ja creada, ens donarà un error dient-nos que aquella alarma

condició ja és una alarma. Seguint aquests passos ja tindrem una Alarma Bàsica creada.

Condición seleccionada

(Debilidad:8,9,10) AND (Dolor:9,10)

Crear Cerrar

Figura 26 – Detall Pantalla Alarmes Bàsiques: Condició Seleccionada

6.1.2.2 Generació d'una Alarma d'Evolució

Les Alarmes d'Evolució eren aquelles que analitzaven la evolució dels Símtomes al llarg del temps o d'una sèrie d'auto avaluacions. Recordem que les alarmes d'evolució també podien ser simples o complexes si les combinem entre elles. En la **Figura 27** podem veure la pantalla de generació de les regles d'Evolució. Per arribar a aquesta pantalla ho farem mitjançant el menú: *Alarmas -> Asignar*

Edición Alarmas de Evolución

Nombre de la alarma: Augment ràpid de la debilitat

Condición

Problema	Variación	Período	Unidad
Sequedad de boca	<	1	2 Días

Agregar

Augment ràpid de la debilitat

Variación Sequedad de boca < 1 por 2 Días

AND

OR

NOT

Eliminar

Importar alarma...

Condición seleccionada

Variación Debilidad > 3 por 2 Evaluaciones

Crear Cerrar

Figura 27 - Pantalla de Generació d'Alarmes d'Evolució

A l'hora de generar les Alarmes d'Evolució la metodologia a seguir és la mateixa que per a les Alarmes Bàsiques: Determinar el nom de l'alarma, generar les condicions a combinar, combinar aquestes condicions a la zona de la pissarra i quan ja tinguem la combinació desitjada, seleccionar-la i crear-la mitjançant el botó de *Crear*.

6.1.3 Assignació d'Alarmes

A l'hora d'assignar les alarmes als agents pacients, no es disposa d'una pantalla diferenciada per a les Alarmes Bàsiques i les Alarmes d'Evolució. Tindrem una pantalla comuna. Aquesta la podem contemplar en la **Figura 28**. Per arribar a aquesta pantalla ho farem mitjançant el menú:

Asignación Alarmas

Agentes Paciente activos

Id Agente	Nombre paciente
11111	Jordi Sabaté

Refrescar Crear Agente Paciente

Detalles Agente Paciente

Id. Agente: 11111 Mostrar

Debilitat Extrema
Augment ràpid de la debilitat

{Debilitad:8,9,10} AND {Dolor:9,10}

Agregar Alarma Eliminar Alarma

Figura 28- Pantalla d'Assignació d'Alarmes

Com es pot apreciar, tenim la pantalla dividida en dues zones principals, en la primera, tindrem un llistat amb els Agents Pacient disponibles. Per cada Agent tindrem el seu

identificador així com el nom del pacient. Des d'aquesta zona també podem donar d'alta nous Agents Pacients si així ho desitgem, per fer-ho disposem del botó *Crear Agente Paciente*. Aquest ens obrirà el diàleg mostrat a la **Figura 17** i el funcionament serà el mateix que l'explicat allí. Si creem nous Agents necessitem fer servir el botó de *Refrescar* per a poder-los veure.

La segona part de la pantalla és on veurem els detalls de l'Agent Pacient seleccionat. En primer lloc se'ns mostrarà l'identificador del pacient (NHC). Si especifiquem un NHC i fem ús del botó *Mostrar* ens mostrarà el pacient escollit en el cas de que aquest existeixi. A continuació tindrem el llistat d'alarmes que té assignat aquest pacient, en aquest llistat hi podem trobar tant Alarmes Bàsiques com Alarmes d'Evolució. En tot moment, per l'alarma que tinguem seleccionada se'ns mostrarà la seva equació.

Per afegir o desassignar alarmes a un pacient tenim els dos botons inferiors. Amb el botó d'*Agregar Alarma* ens apareixerà un menú emergent (**Figura 29**) que ens permetrà escollir si volem afegir una Alarma Bàsica o un Alarma d'Evolució, a l'escollir el tipus d'alarma se'ns obrirà una pantalla com la que teníem per importar alarmes quan les estàvem generant (**Figura 25**). Per a eliminar una alarma, solament l'hem de seleccionar i clicar sobre el botó *Eliminar Alarma*.



Figura 29 - Menú Afegir Alarma

6.1.4 Alarmes Actives

Quan una Alarma s'activi, l'Agent Pacient que ho hagi detectat enviarà una notificació a l'Agent Metge corresponent. Quan l'Agent Metge rebí la notificació passarà a mostrar un missatge d'alerta conforme s'ha rebut una Alarma (**Figura 30**).

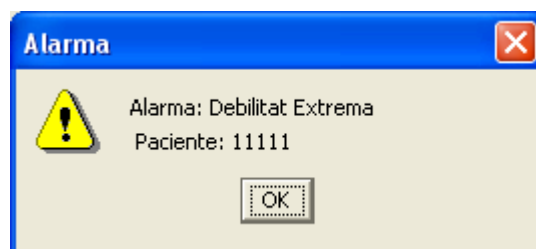


Figura 30 - Avís Alarma Activada

Al mateix temps, el comptador d'alarmes actives s'incrementarà (**Figura 31**).

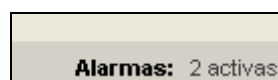


Figura 31 - Comptador Alarmes Actives

Si cliquem sobre el comptador d'alarmes actives se'ns obrirà una pantalla amb les Alarmes que actualment estan *Actives*, pendents de passar a *Històriques*. Aquesta pantalla és la que podem veure a la **Figura 32**.

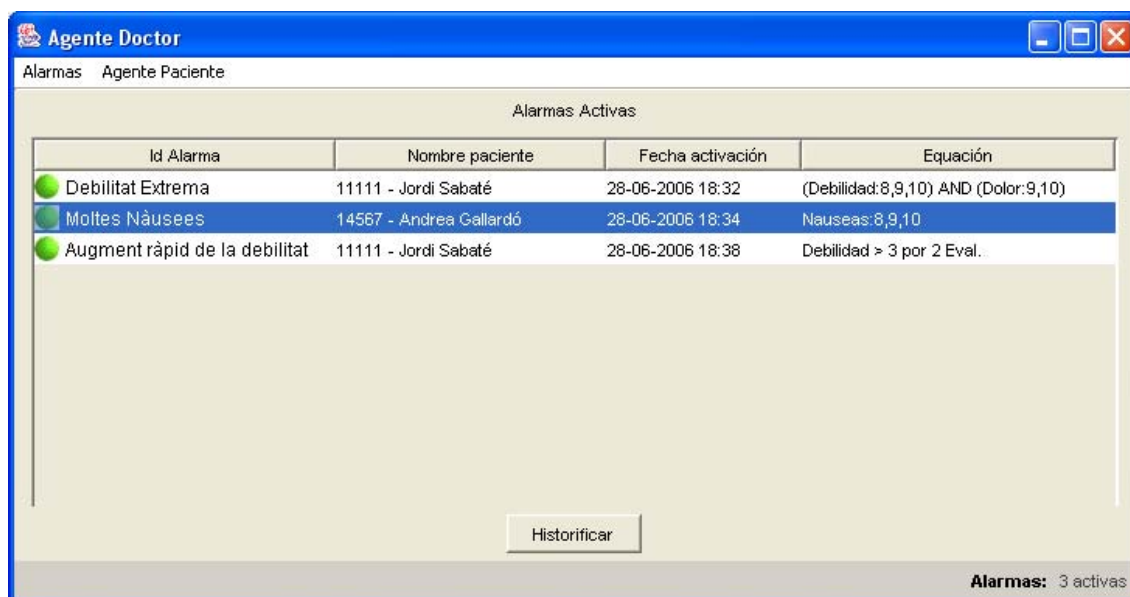


Figura 32 - Pantalla Alarmes Actives

En aquesta pantalla podem obtenir la informació de les alarmes que estan actives en aquest moment, les dades que es mostren per a cada alarma són *l'Identificador de l'Alarma, el NHC i el Nom del Pacient, la data i hora d'activació i l'Equació de l'Alarma*. Per a Historificar una alarma l'hem de seleccionar i clicar el botó d'*Historificar*.

Per a accedir a aquesta pantalla també ho podem fer a través del menú: *Alarmas -> Alarmas Activas*

6.1.5 Historial d'Alarmes

A l'Historial d'Alarmes hi podem consultar tant les alarmes que estan actives com les històriques. Per a accedir-hi ho farem mitjançant la opció de menú: *Alarmas -> Historial de Alarmas*. La Pantalla que se'ns mostrarà serà com la que podem veure a la **Figura 33**.

Com en el cas de les Alarmes Actives la informació que n'obtindrem serà la mateixa: *l'Identificador de l'Alarma, el NHC i el Nom del Pacient, la data i hora d'activació i l'Equació de l'Alarma*. En aquest cas, per a poder distingir si una alarma està activa o és històrica ens fixarem en el punt, si és verd voldrà dir que est tracta d'una alarma activa, mentre que si està en vermell significarà que es tracta d'una alarma historificada.

Com s'ha comentat anteriorment les Alarmes que es mostraran en aquest punt seran les dels dos últims mesos. Si es volen mostrar alarmes d'altres franges temporals, ho farem

mitjançant els camps de Data d'Inici i Data de fi que apareixen en la pantalla i clicant sobre el botó *Aceptar*.

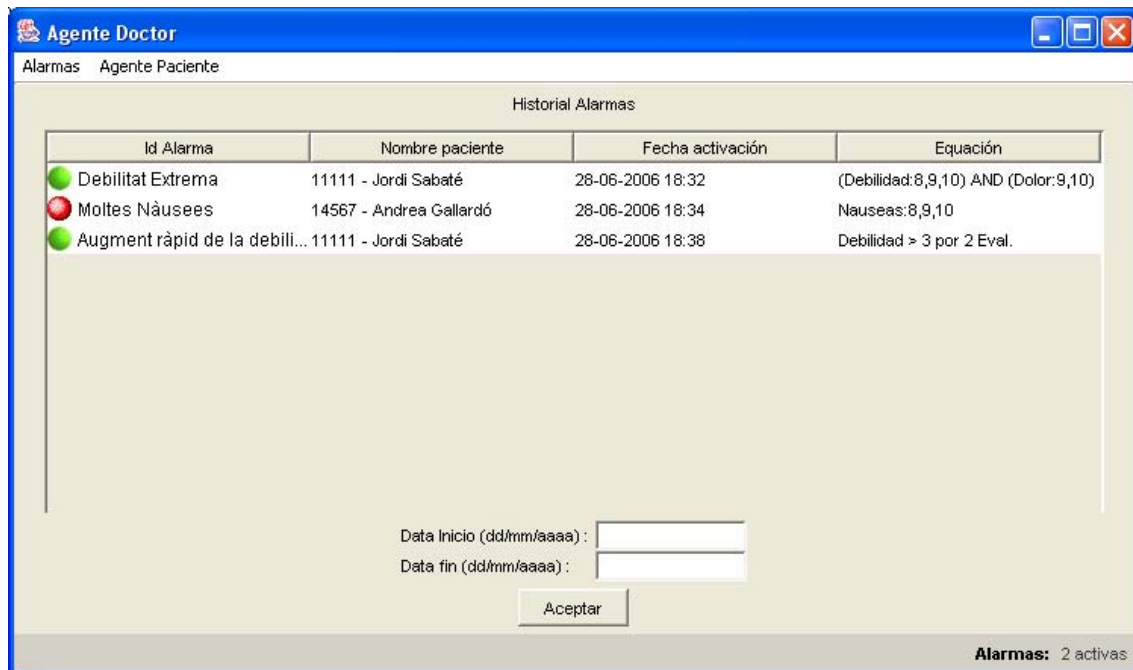


Figura 33 - Historial Alarmes

6.2 Agent Estadístic

L'Agent Estadístic proporciona tres funcionalitats, la de generació d'estadístiques, la de generació de Gràfiques i la definició d'alarmes que seran accessibles per tots els agents Metge. A l'hora d'accedir a l'Agent Estadístic també ens trobarem una pantalla de validació en la que ens demanarà el password de l'administrador (**Figura 34**). El funcionament serà el mateix que per l'Agent Metge.



Figura 34 - Validació Administrador

Un cop haguem validat el password ja podrem fer ús de l'Agent Estadístic, la pantalla del qual podem veure a la **Figura 35**.

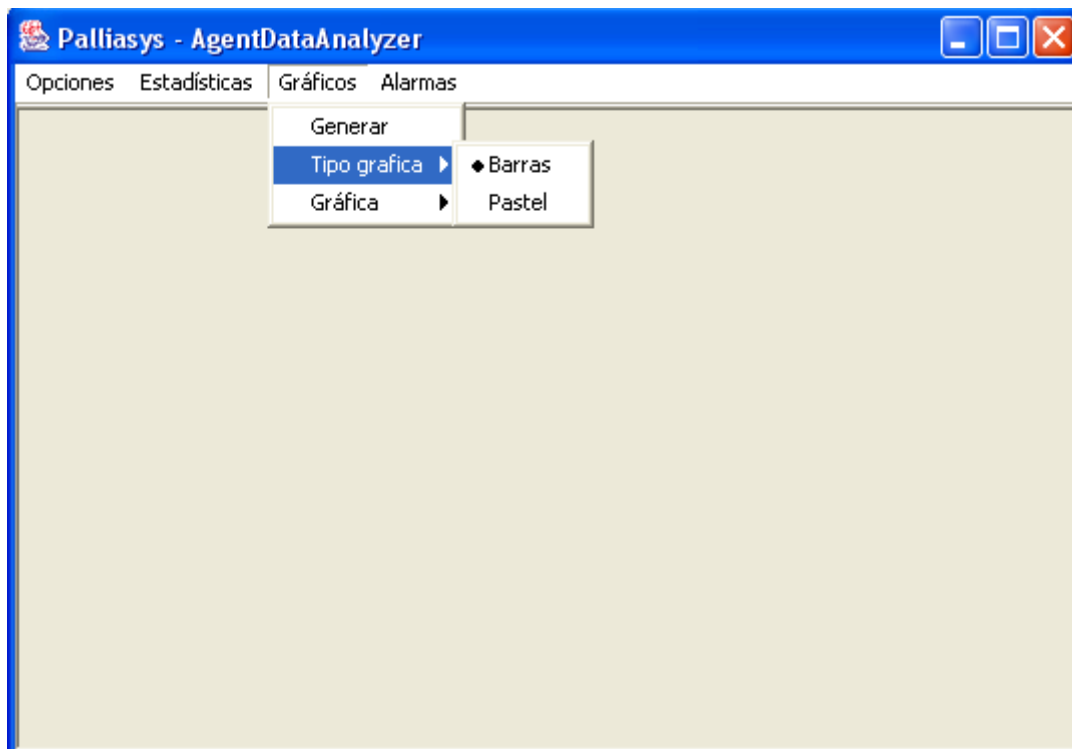


Figura 35 - Agent Estadístic

L'Agent Estadístic ens proporciona quatre opcions de menú amb els seus sub-menús corresponents.

Opciones: És on tenim les opcions generals tant que són vàlides tant per les estadístiques com per les gràfiques.

- **Período:** Ens obrirà la finestra on ens permetrà especificar el període per el que volem obtenir les gràfiques i les Estadístiques (**Figura 36**)
- **Guardar:** Ens obrirà una finestra típica d'exploració per a guardar l'estadística o la gràfica que tinguem activa. Les estadístiques es guardaran com un fitxer *.html* i les gràfiques com un *.jpg*.

Estadísticas: Des d'aquest menú podrem escollir les dades que vulguem en les estadístiques així com generar-les.

- **Generar:** Ens generarà l'estadística per les dades i el període que haguem especificat.
- **Datos:** Ens obrirà la finestra on podrem escollir les dades que ens entraran a l'estadística.

Gráficos: Ens permetrà generar les diferents gràfiques.

- **Generar:** Ens generarà la gràfica per el període, el tipus i la gràfica en concret que haguem escollit.

- **Tipo Gráfica:** Ens permetrà escollir el tipus de gràfica que vulguem (Pastís Barres)
- **Gráfica:** Des d'aquí podrem escollir la gràfica que desitgem (distribució de sexes, tipus de càncer, etc.)

Alarmas: Des d'aquest menú podrem definir les alarmes generals.

- **Editar Alarma Bàsica:** Ens obrirà la finestra de disseny de les Alarmes Bàsiques que ens permetrà generar noves Alarmes Bàsiques o recombinar les que ja tinguem.
- **Editar Alarma de Evolución:** Ens obrirà la finestra corresponent al disseny de les Alarmes d'Evolució. A l'igual que l'anterior, podrem crear-ne de noves o recombinar les que hàgim definit prèviament.

6.2.1 Opcions generals

Les opcions generals seràn vàlides tant per les estadístiques com per les gràfiques.

6.2.1.1 Definir el Període

Amb aquesta pantalla (**Figura 36**) podrem definir el període per el que volem mostrar les Estadístiques o les Gràfiques.

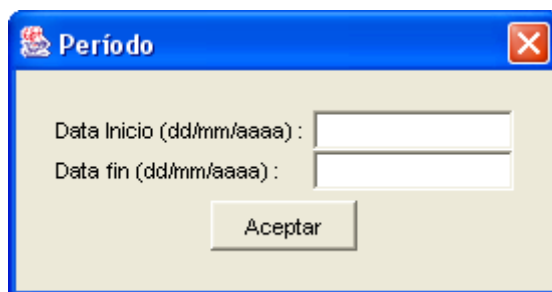


Figura 36 - Pantalla de definició del període

6.2.1.2 Guardar Estadístiques i Gràfiques

Un cop s'hagi generat una estadística o una gràfica (en els següents apartats es veurà com fer-ho), es tindrà l'opció de guardar-les a disc. Les estadístiques es podran guardar en format html i les gràfiques es guardaran en format jpg.

6.2.2 Estadístiques

Per a generar les estadístiques primer de tot serà necessari definir quines ens volen generar. Per a fer-ho es disposa de l'arbre amb caselles seleccionables que es pot veure a la **Figura 37**.

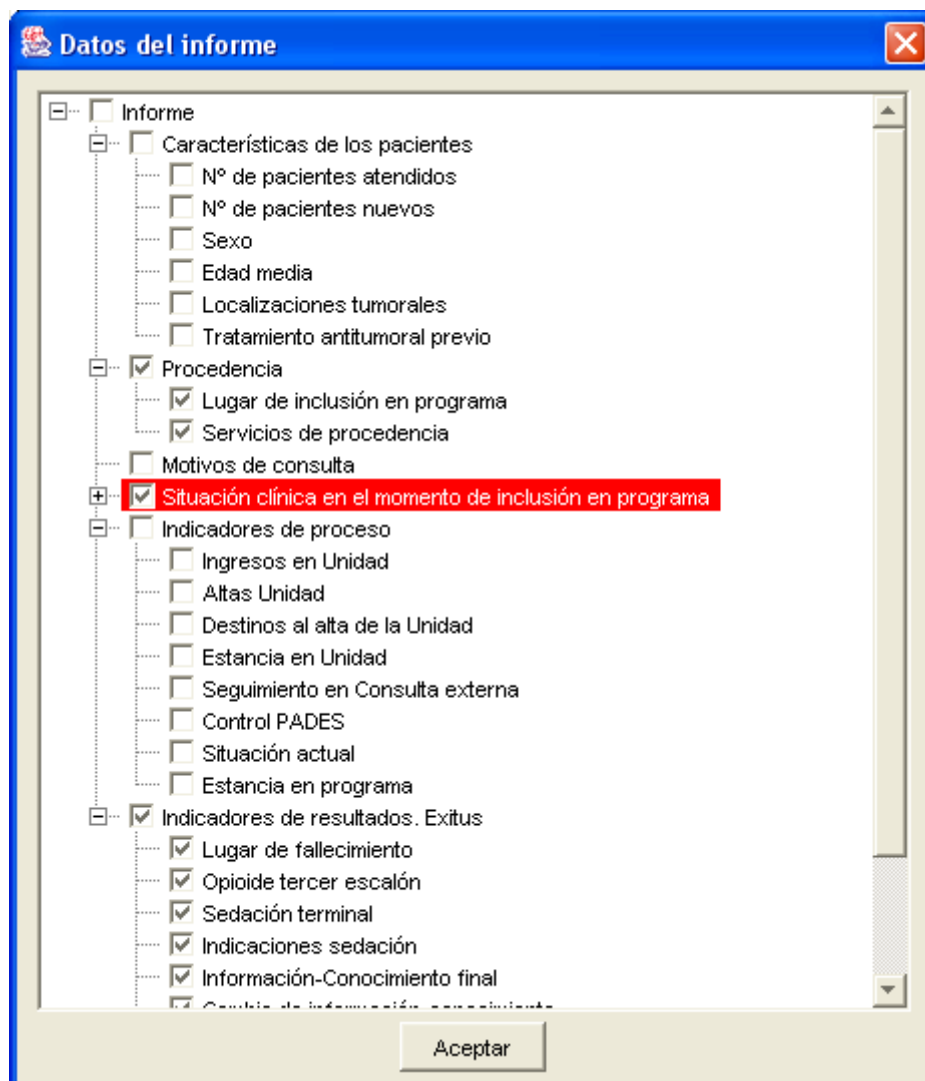


Figura 37 - Dades a seleccionar per a mostrar a l'Informe

Un cop haguem seleccionat les dades estadístiques desitjades, les generarem amb l'opció de menú: *Estadísticas -> Generar*

L'informe que generarà pot variar de contingut depenent de les opcions seleccionades amb la pantalla anterior. En el cas de que es seleccionessin totes les opcions possibles les dades que es mostrarien serien les següents en l'ordre mostrat a continuació.

CARACTERÍSTICAS DE LOS PACIENTES

Nº pacientes atendidos:

Nº de pacientes nuevos:

Sexo:

Varones:

Hembras:

Edad media:

Localizaciones tumorales:

Pulmón:

Colorectal:

O. Desconocido
Estómago
Mama:
Vejiga:
ORL
Páncreas:
Próstata:
Linfoma:
Hepatocarcinoma
Renal:
Leucemia aguda:
Mesotelioma
Ovario:
Vía biliar:
Cérvix:
Glioblastoma
Sarcoma
Esófago:
Endometrio:
Otros

Tratamiento antitumoral previo (fase avanzada)

SI:
NO:

PROCEDENCIA

Lugar de inclusión en programa:

Interconsulta hospitalaria:
Consulta externa:

Servicios de procedencia:

Oncología:
Consulta externa:
Interconsulta:
Otros Servicios:
Medicina Interna:
Cirugía:
Urgencias
Neumología
Hematología
Digestivo:
Urología
Traumatología
Neurocirugía
Geriatría
Ginecología
PADES

MOTIVOS DE CONSULTA

(pueden existir varios en un mismo paciente)

Síntomas:

Otros motivos:

Valoración y seguimiento:
Planificación alta: -
Valorar traslado

SITUACIÓN CLÍNICA EN EL MOMENTO DE INCLUSIÓN EN PROGRAMA

Performance status

1 (síntomas, act normal)
2 (<50% en cama):

3:(>50% en cama): 1

4: encamado:

Molestia prioritaria (subjetiva)

Preocupación prioritaria (subjetiva)

Presencia de dolor:

leve

moderado

severo

irresistible

Reciben opioide de tercer escalón:

Morfina

Fentanilo

Dolantina

Metadona

INFORMACIÓN-CONOCIMIENTO DE LA SITUACIÓN

Ninguno

Dudoso

Conoce diagnóstico

Posibilidad de morir

Total

INDICADORES DE PROCESO.

INGRESOS EN UNIDAD

episodios:

pacientes:

ALTAS UNIDAD

pacientes

Destinos al alta de la Unidad:

Domicilio:

C. Sociosanitario:

Estancia en Unidad:

Media:

Mediana:

Seguimiento en Consulta externa:

Control PADES:

Situación actual (a fecha)

activos:

pérdida seguimiento

exitus

Estancia en programa (a fecha).

Exitus (n=395)

Media:

Mediana:

INDICADORES DE RESULTADOS. EXITUS, N= 395

LUGAR DE FALLECIMIENTO

domicilio:

Unidad:

Otros Servicios H Sant Pau:

Urgencias:

CSS:

OPIOIDE TERCER ESCALÓN:

morfina:

fentanilo:

metadona:

dolantina:

SEDACIÓN TERMINAL

unidad:
otros servicios:
domicilio:

Indicaciones sedación:

Angustia-agitación
Disnea
Dolor
Mal estado general
Crisis convulsiva
Petición enfermo
Abdomen agudo
Desconocido

INFORMACIÓN-CONOCIMIENTO FINAL

ninguno
dudoso
conoce diagnóstico
posibilidad de morir
total

Cambio de información-conocimiento:

Petición eutanasia:

Ocasional
Permanente

Expresión deseos de muerte:

Ocasional
Permanente
Aceptación:

Serenidad familiar

escasa:
bastante:
mucha:

SEGUIMIENTO EN DUELO:

No
Si

6.2.3 Gràfiques

Disposen de dos tipus de gràfiques, les de barres i les de pastís. Qualsevol gràfica pot generar-se en els dos formats. Per a generar la gràfica desitjada Les gràfiques es generaran d'una en una.. A l'hora de generar una gràfica primer s'ha d'escollir el tipus a través del menú *Gràficos->Tipo Gráfica*, després s'ha d'escollir el model de gràfica desitjat. Finalment s'ha de Generar. En l'exemple de la **Figura 38** podem veure com s'ha escollit un tipus de barres i la gràfica és per Motius de Consulta. En la **Figura 39** podem veure la mateixa gràfica, però el format en aquest cas és de Pastís.

Les gràfiques ens permeten tenir una visió ràpida de la comparativa de dades. Totes les gràfiques es poden guardar en format **jpg**.

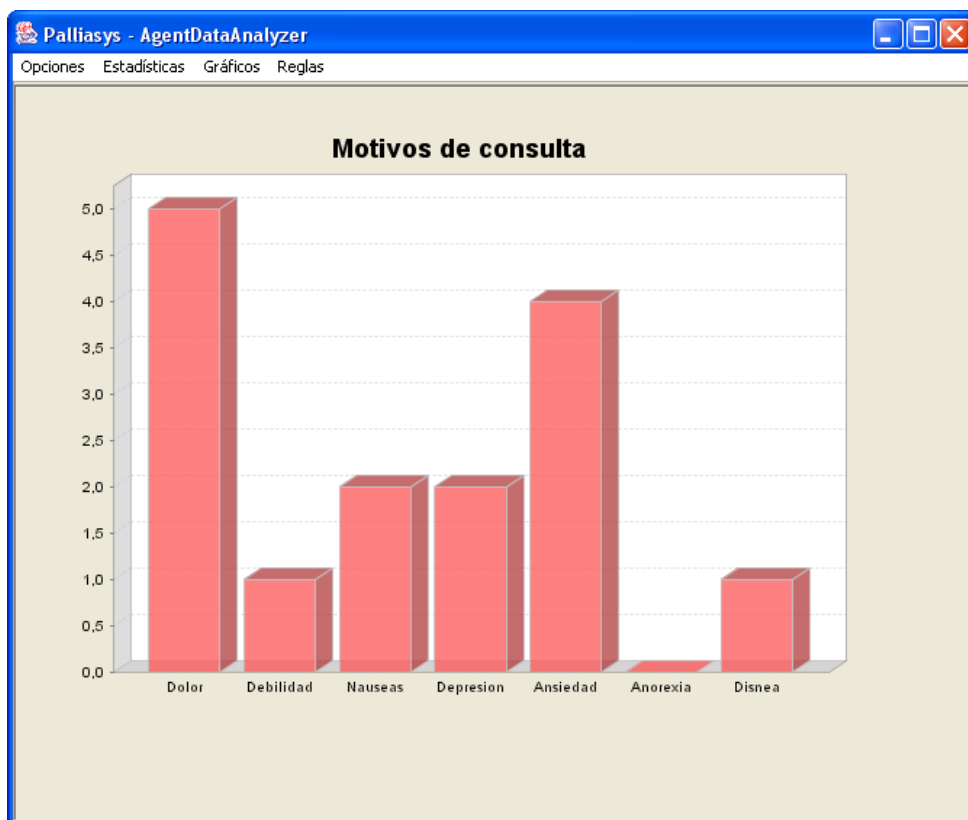


Figura 38 - Exemple de Gràfica de Barres



Figura 39 - Exemple de Gràfica de Pastís

6.2.4 Alarmes

La definició d'alarmes serà com en l'Agent Metge. La diferència principal resideix en que l'Administrador no assignarà les alarmes a cap Agent Pacient ja que no en té cap d'associat. En canvi, les alarmes que defineixi seran visibles per part de tots els Agents Metge.

7 Joc de Proves

Per tal de comprovar el correcte funcionament del projecte que ens ocupa, s'ha realitzat un exhaustiu joc de proves. Aquí mostrarem els casos més rellevants. Comencem amb les Validacions d'Accés:

7.1 Validacions d'Accés

Provarem primer per l'Agent Metge. El primer cas serà per amb un password incorrecte. En segon lloc ho farem però amb el password correcte.

- 1) Engeguem l'aplicació mitjançant el fitxer *agentMetge.bat*. Tot funciona correctament i ens apareix la pantalla de validació.
- 2) Introduïm un password incorrecte (234324) i ens salta el missatge d'error tal i com era previst.
- 3) Després d'Acceptar el missatge d'error tornem a intentar a entrar al sistema amb un password correcte aquest cop. I tot funciona com era previst, ja podem treballar amb l'Agent metge.

Ara ho provarem per a un Administrador, ho farem igual que per l'Agent Metge, primer per un cas correcte i després per un cas Incorrecte.

- 1) Engeguem l'aplicació mitjançant el fitxer *agentEstadistic.bat*. Tot funciona correctament i ens apareix la pantalla de validació.
- 2) Introduïm un password incorrecte (234324) i ens salta el missatge d'error tal i com era previst.
- 3) Després d'Acceptar el missatge d'error tornem a intentar a entrar al sistema amb un password correcte aquest cop. I tot funciona com era previst, ja podem treballar amb l'Agent Estadistic.

Ara que ja hem validat que l'accés es realitza de forma correcta, passem a provar les diferents funcionalitats del sistema.

7.2 Alta dels Agents Pacient

En aquest apartat ens centrarem en la gestió dels Agents Pacient on, primer ho provarem per el cas d'un NHC d'un pacient inexistent, després provarem per un pacient existent a continuació tornarem a provar de donar d'alta l'agent que acabem de donar d'alta correctament des del mateix Agent Metge i des d'un de diferent.

- 1) Accedim a la pantalla d'Alta d'Agent Pacient i introduïm un NHC inexistent (4234234555), ens salta l'error que aquell pacient no existeix a l'intentar donar d'alta l'Agent.

- 2) Realitzem la mateixa operació, aquest cop, però amb NHC d'un pacient que sí que existeix. Aquest cop sí que es crea l'Agent
- 3) Tornem a provar de donar d'alta un Agent Pacient amb el mateix NHC, ens salta un error de que ja tenim definit aquest agent
- 4) Engueguem un altre Agent Metge i des d'ell intentem donar d'alta el mateix pacient. Ens saltarà l'error de que aquell Agent Pacient ja està creat i per un altre Agent Metge.

7.3 Alarmes

Farem el seguiment d'un cicle de vida complet d'una Alarma Bàsica (també s'ha fet per les Alarmes d'Evolució, però no s'exposa perquè seria repetir el mateix)

- 1) Primer de tot anirem a la pantalla de generació de les Alarmes Bàsiques i introduïm el nom desitjat de l'Alarma: *Debilitat Extrema*
- 2) Generem un parell de condicions seguint la metodologia explicada en el tutorial.
- 3) Combinem aquestes dues condicions mitjançant un OR. Comprovem que si no seleccionem a les dues ens dona un missatge d'error.
- 4) Seleccionem el resultat i pitgem a crear alarma, com que ja existeix una alarma amb el mateix nom (que havíem creat prèviament), ens donarà un error i ens demanarà que li canviem el nom. L'anomenarem doncs, *Debilitat Extrema 2*.
- 5) Un cop generada l'alarma, anem a la pantalla d'Assignacions, escollim l'Agent Pacient que hem donat d'alta abans i procedim a assignar-li l'alarma tal i com es comenta en el tutorial
- 6) Ara que ja tenim l'Alarma generada i assignada volem comprovar que tot vagi bé. Arranquem el navegador, obrim la pàgina inicial de l'aplicatiu web, ens identifiquem com el pacient al que hem associat l'alarma i generem una auto avaluació que activarà l'alarma.
- 7) Immediatament després de fer això, ens arriba un avís al nostre agent metge. L'Alarma també ens surt al llistat d'alarmes actives i a l'Historial.
- 8) A continuació donem de baixa l'Agent Pacient que hem creat (seguint com sempre la metodologia esmentada en el tutorial. De pas, també intentem donar de baixa Agents Pacients que no existeixen i ens dona un error.
- 9) Tornem a seguir els mateixos passos que abans per a tornar a generar una Auto avaluació que ens activaria l'Alarma en el cas de que no haguéssim donat de baixa l'Agent pacient i, com era d'esperar, no ens arriba cap Alarma a l'Agent Metge.

7.4 Dades Estadístiques

7.4.1 Estadístiques en HTML

Per a realitzar les proves de l'Agent Estadístic primer escollirem un període que comprendrà l'últim mes, a continuació escollim quines són les dades desitjades. Les escollim totes i generem el document. Tot surt com era d'esperar. Surt tot el llistat comentat a la secció del tutorial amb dades que si es contrasten amb la Base de Dades es comprova que són correctes.

7.4.2 Gràfiques

Ens quedem amb el període que havíem definit per les estadístiques, en primer lloc, escollirem una Gràfica de pastís i en segon lloc una de Barres. Són les que hem pogut veure en la secció del tutorials.

7.5 Seguretat - Fitxers de Log

Com ja hem comentat anteriorment, en aquest projecte disposarem de tres fitxers de log, en cadascun s'hi emmagatzemarà les dades corresponents. Després de realitzar tots els jocs de proves dels apartats anteriors, el contingut resultant dels fitxers hauria de ser de 2 Accessos Correctes, 2 Accessos incorrectes i 27 Accessos a dades. El contingut obtingut en els fitxers que és l'esperat.

8 Conclusions i Treball Futur

8.1 Conclusions

En primer lloc voldria agrair el suport obtingut per part del meu cap de projecte, el Dr. Antonio Moreno, així com dels companys que també han treballat amb *PalliaSys*. A banda, també m'agradaria esmentar la col·laboració prestada pel personal de la Unitat de Cures Pal·liatives de l'Hospital de la Santa Creu i Sant Pau.

Un dels punts més interessants en aquest projecte ha estat que forma part del projecte *Palliasys*. Aquest fet ha provocat que s'hagi hagut d'integrar amb altres projectes incrementant l'interès i fent també que a banda de tocar les tecnologies que tractava en el meu projecte també en toqués més indirectament d'altres. A més ha estat interessant la integració del Sistema Multi Agent amb la tecnologia PHP.

El fet de que el projecte *Palliasys*, en el que està englobat aquest projecte, estigui plantejat com a un sistema a implantar fa que tingui un interès més elevat ja que d'aquesta forma es podrà comprovar com es comporta el Sistema Multi Agent definit en el món real.

Amb aquest projecte, a més, hem pogut explotar els avantatges que proporciona el fet de fer servir un **SMA** enfront de la programació tradicional per a problemes com el plantejat en un cas com el de *PalliaSys*. Aquest projecte proporciona un sistema d'Alarmes molt eficaç i precís que permet un control molt bo i innovador de l'estat i evolució dels pacients en la Unitat de Cures Pal·liatives. A banda, també proporciona altres funcionalitats no tan innovadores, però igualment útils com la de la generació d'estadístiques i gràfiques.

8.2 Treball Futur

Com a treball futur es podrien considerar unes quantes ampliacions i modificacions del *Sistema Multi Agent*. Uns dels casos que podríem destacar serien els següents:

- **Agent Pacient assignat a més d'un Agent Metge:** Actualment, un Agent Pacient està assignat a un sol Agent Metge ja que va ser com es va plantejar des d'un principi, en un futur, però es podria plantejar que un Agent Pacient es pogués assignar a més d'un Agent Metge.
- **Assignació d'alarmes globals:** En aquest moment, es poden definir alarmes de forma global, però no així assignar-les de forma global, és a dir, a tots els Agents Pacient actius. Una millora, doncs, seria la de possibilitar l'assignació d'aquestes alarmes definides de forma global per part de l'Agent Analitzador.
- **Més informació associada a les Alarmes:** Quan veiem la informació continguda en les alarmes, aquestes mostren la informació referent al pacient que les ha

activat, l'hora i dia d'activació i l'equació i identificador de l'Alarma que s'ha activat. També podria ser interessant mostrar la informació de la auto avaluació que l'ha activat, en el cas de les Alarmes Bàsiques, o l'evolució que han sofert els valors que l'han activat en el cas de les Alarmes Incrementals.

S'ha de tenir en compte que algunes funcionalitats que es podrien considerar interessants i que no s'han inclòs en aquest projecte, com per exemple l'estudi de les dades obtingudes, les podem trobar al projecte del Raul Mateos [2].

Annex A Implementació d'un SMA en JADE

JADE (*Java Agent Development Enviroment*) [5] és un suport per a desenvolupar *Sistemes Multi Agent*. Està format per un conjunt de llibreries escrites en Java que ens permeten el poder desenvolupar Sistemes Multi Agent més fàcilment.

El que ens permet fer **JADE** [5] és desenvolupar un SMA sense que ens haguem de preocupar de com resollem alguns detalls de baix nivell. A més, **JADE** també ens ofereix un entorn d'execució per als nostres agents.

A continuació descriurem les parts de **JADE** [5], com s'implementen agents en aquest entorn i veurem per sobre les eines gràfiques que ens dóna.

A.1 Plataforma d'Agents de JADE

Per JADE, els agents resideixen en un entorn predefinit anomenat **plataforma**. Cada plataforma està dividida en contenidors i cadascun d'ells és on es trobaran continguts els agents. Els contenidors poden entendre's com *dominis d'agents*.

La plataforma s'engega en una màquina determinada. En canvi, els agents podran trobar-se executant en d'altres estacions o en la mateixa on s'ha engegat la plataforma. Aquesta plataforma permet als agents comunicar-se emprant el **FIPA-ACL**, esmentat anteriorment.

Tal com es defineix a la FIPA, en cada plataforma hi ha d'haver dos agents que tenen unes funcions imprescindibles per al funcionament del sistema: l'**Agent DF (Directory Facilitator)** i l'**Agent AMS (Agent Management System)**. Tots dos comentats en la secció dels Sistemes Multi Agent.

Aquests dos agents són d'existència obligatòria. Quan ens trobem en el cas de que un agent A desitja enviar un missatge a un agent B el qual desenvolupa la tasca T, l'agent A preguntarà al DF quin o quins agents poder fer-se càrrec de la tasca T. Llavors el DF li retornarà l'adreça de l'agent B. Quan es vol conèixer l'adreça d'un altre agent, la pregunta es fa al AMS.

A.2 Implementació d'Agents

Un agent es crea mitjançant l'extensió de la classe **Agent**. Un cop hem estès la classe ens cal implementar el mètode *setup()* que és obligatori. Aquest s'executa el primer cop que l'agent entra en el sistema. S'acostuma a usar per a realitzar les inicialitzacions pertinents i posar en funcionament algun comportament per a iniciar les tasques de l'agent.

Després de crear l'esquelet i implementar el *setup()*, cal implementar els serveis de l'agent. Per a fer-ho utilitzarem els **behaviours** (o comportaments). És obligatori implementar com a mínim un *behaviour* per agent.

Aquests *behaviours* són els que controlen les seves accions. Un *scheduler* intern serà l'encarregat d'anar controlant automàticament l'execució de tots els comportaments.

JADE ens proporciona una col·lecció bàsica de *behaviours* que ens permeten implementar diferents protocols de comunicació. Tenim dos grans grups, els simples i els complexos. Els comportaments simples són els que no tindran fills. En canvi, els complexos poden tenir **subBehaviours** associats.

Normalment, a l'hora de fer servir els *behaviours* el que es fa és estendre el *behaviour* desitjat i rescirure els mètodes. A l'apartat de disseny, quan expliquem els agents, també comentarem breument el comportament dels *behaviours* que estenen.

A.3 Els paquets de JADE

Com hem esmentat abans, JADE està compost per una sèrie de paquets escrits en JAVA. Els més importants són els que llistem a continuació:

- *jade*: En aquest paquet és on es troben localitzades les classes que es fan servir per a engegar el Jade.
- *jade.core*: És el nucli del sistema. Proporciona la classe *jade.core.Agent* que és imprescindible ja que és de la que deriven tots els agents. A més, també conté el paquet *jade.core.behaviour* dins del qual s'hi troben les classes de comportaments suportats.
- *jade.lang*: Conté les classes específiques per a suportar un llenguatge de comunicació entre agents (ACL). Dintre d'ell hi ha el paquet *jade.lang.acl*.
- *jade.domain*: Conté totes les classes per representar la plataforma d'agents FIPA i els models del domini, tals com entitats per a l'administració d'agents, llenguatges i ontologies (AMS, DF, ACC ...).
- *jade.proto*: Paquet que conté els protocols estàndard d'interacció entre agents definits per la FIPA.
- *jade.gui*: Aquest paquet és el que ens permetrà que els nostres agents tinguin una part gràfica.

A.4 Entorn gràfic del JADE

A més de tot el que hem explicat, el JADE també ens proporciona un entorn gràfic on podem manipular el nostre SMA fàcilment. Traient o posant agents dinàmicament, creant-ne de nous, etc.

Aquest entorn s'engega activant una opció a la línia de comandes. Així, si posem la opció `-gui` se'ns engegarà l'entorn gràfic del JADE on podrem veure el funcionament

del nostre sistema. Els missatges que s'envien entre els agents, el que reben, etc. Tot això gràcies a que el sistema posseeix un *Sniffer* capaç de fer-ho. En la **Figura X** podem veure una visió general del **JADE Remote Agent Management GUI**.

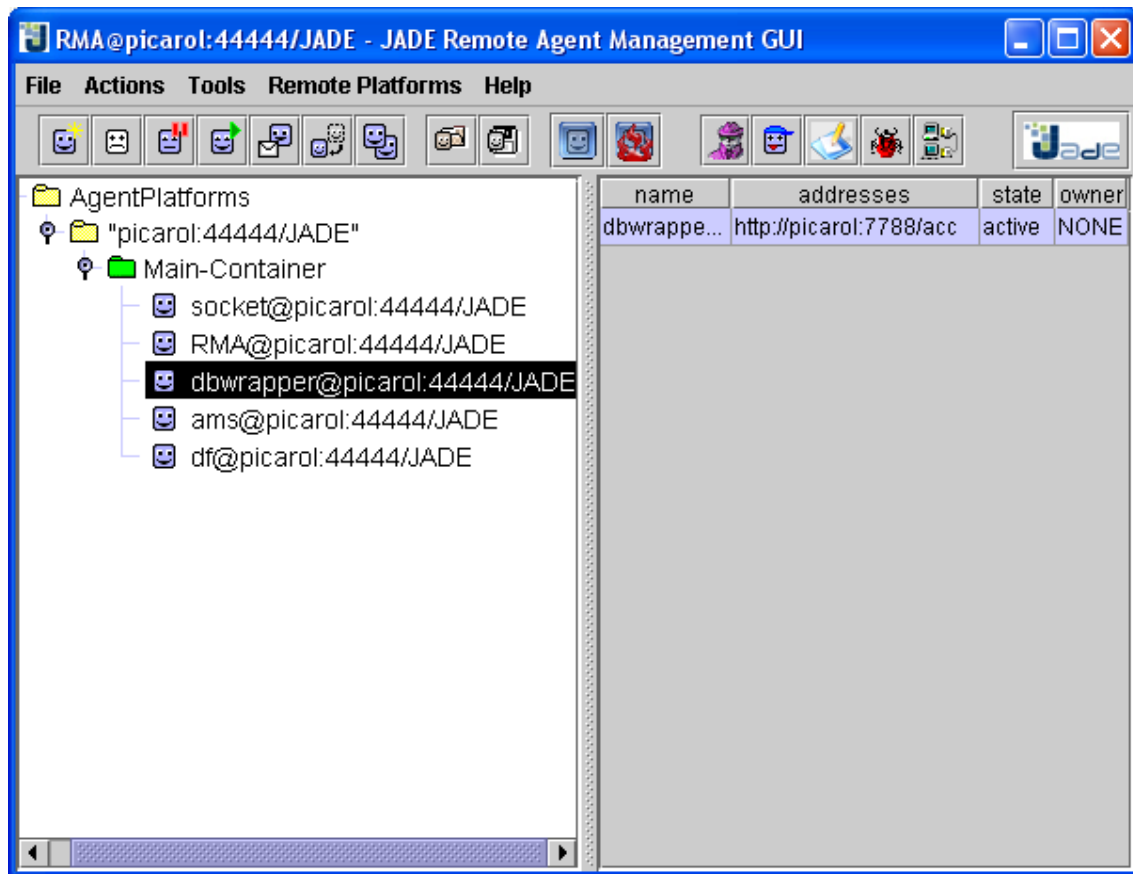


Figura 40 - Jade Remote Agent Management GUI

Referències i Bibliografia

- [1] Projecte PalliaSys: Uso de las nuevas tecnologías de la información y las comunicaciones para facilitar el tratamiento de pacientes paliativos. A.Moreno, D.Riaño, A.Valls
- [2] Raul Mateos: Sistema de análisis de datos médicos según itinerarios y evoluciones. Escola Tècnica Superior d'Enginyeria, Universitat Rovira i Virgili. Juny 2005.
- [3] García Rué, P. Sistema telemático para la asistencia médica y la ayuda a la toma de decisiones en una unidad de curas paliativas. Proyecto Final de Carrera de Ingeniería Informática (Director: Dr. David Riaño). Escola Tècnica Superior d'Enginyeria, Universitat Rovira i Virgili. Juny 2003.
- [4] Especificacions de la FIPA (Foundation for Intelligent Physical Agents). <http://www.fipa.org/specs/fipa00061/>
- [5] JADE (Java Agent DEvelopment framework) <http://jade.cselt.it>.
- [6] Foundation for Intelligent Physical Agents (6/12/2002). *FIPA ACL Message Structure Specification* (SC00061). FIPA.
- [7] Foundation for Intelligent Physical Agents (6/12/2002). *FIPA SL Content Language Specification* (SC00008). FIPA.
- [8] Foundation for Intelligent Physical Agents (6/12/2002). *FIPA Agent Management Specification* (SC00023). FIPA.
- [9] Foundation for Intelligent Physical Agents (6/12/2002). *FIPA Request Interaction Protocol Specification* (SC00026). FIPA.