

Universidad Politécnica de Madrid  
Escuela Técnica Superior de Ingenieros de Telecomunicación



# **CASO DE ESTUDIO DE AGENTES MÓVILES EN EL HOGAR DIGITAL**

**TRABAJO FIN DE MÁSTER**

**Rubén González Criado**

2011



Universidad Politécnica de Madrid  
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en  
Ingeniería de Redes y Servicios Telemáticos**

**TRABAJO FIN DE MÁSTER**

**CASO DE ESTUDIO DE AGENTES  
MÓVILES EN EL HOGAR DIGITAL**

Autor  
**Rubén González Criado**

Director  
**Ignacio Soto Campos**

Departamento de Ingeniería de Sistemas Telemáticos

2011

## Resumen

El objetivo fundamental de este proyecto consiste en estudiar la viabilidad de la tecnología de agentes móviles en el hogar digital. Para ello, se ha llevado a cabo una exhaustiva búsqueda de información en la web relacionada tanto con el estado actual del hogar digital como con la tecnología de agentes.

En el estudio de la tecnología de agentes, en primer lugar, se ha buscado información sobre sus características, funcionalidad, arquitectura, plataformas de desarrollo, etc. que ha permitido conocer en detalle todo lo relacionado con dicha tecnología. Posteriormente, se ha analizado diferentes estándares de agentes móviles, como son MASIF y FIPA, para obtener una idea precisa de cómo funciona dicha tecnología a bajo nivel (estructura, servicios ofrecidos, interfaz de programación, etc.).

Además, dado el gran interés investigador que existe en la actualidad sobre la tecnología de agentes en entornos inalámbricos, se ha estudiado un conjunto de soluciones arquitecturales para este tipo de escenarios, como son Agilla e In-Motes.

Partiendo del trabajo realizado hasta este momento, se ha elaborado un modelo de agentes en el hogar digital, que en definitiva, es otro de los objetivos importantes. En dicho modelo, se da cabida a la tecnología de agentes enumerando y describiendo los diferentes agentes existentes para cubrir una serie de servicios ofrecidos en el hogar digital.

Una vez realizado este nexo de unión entre la tecnología de agentes y el hogar digital, se ha dado un paso más allá, especificando una plataforma (simula a grandes rasgos el comportamiento de una pasarela) de comunicación entre las diferentes tecnologías implantadas en el hogar digital.

Para finalizar, se expone brevemente como sería el funcionamiento de un servicio concreto en el hogar digital, junto con las tecnologías estudiadas y dentro del escenario propuesto.



## Abstract

The purpose of this project is to study the feasibility of mobile agent technology in the digital home. It has conducted an exhaustive search for information on the web related to both the current state of the digital home with the agent technology.

For the study of the technology of agents, on the first place, it has sought information about its features, functionality, architecture, development's platform, etc it has revealed in detail everything about the technology.

Subsequently, we analyzed different standards of mobile agents, such as MASIF and FIPA to get an accurate idea of how this technology at a low level (structure, services offered, programming interface, etc.). Moreover, given the great research interest currently exists on agent technology in wireless environments, we have studied a set of architectural solutions for these scenarios, such as Agilla and In-Motes.

Based on the work done so far, has developed a model of agents in the digital home, in short, is another important objective. In this model, it allows for the listing agent technology and describes the different existing agents to cover a range of services offered in the digital home.

Once this link between technology players and digital home is done, one step further it's being taken by specifying a platform (roughly simulates the behavior of a gateway) communication between the various technologies implemented in the digital home.

Finally, we briefly as would be the performance of a particular service in the digital home, along with the technologies studied and within the proposed scenario.



## Índice general

Resumen .....	i
Abstract.....	iii
Índice general .....	v
Índice de figuras.....	vii
Índice de tablas.....	ix
1 Introducción.....	1
2 Estado del arte .....	3
2.1 Tecnologías en el hogar digital .....	3
2.1.1 KNX.....	3
2.1.2 LonWorks .....	6
2.1.3 ZigBee.....	7
2.1.4 DLNA.....	8
2.1.5 uPnP.....	10
2.1.6 UWB.....	12
2.1.7 Conclusiones sobre las distintas tecnologías en uso en el HD.....	15
2.2 Tecnología de agentes.....	17
2.2.1 Introducción a los agentes .....	17
2.2.2 Definición de agente .....	18
2.2.3 Características de un agente .....	19
2.2.4 Clasificación de agentes.....	21
2.3 Agentes móviles.....	25
2.3.1 Conceptos de agentes móviles .....	25
2.3.2 Tecnología de los agentes móviles.....	28
2.3.3 Ventajas de los agentes móviles.....	31
2.3.4 Aplicaciones de los agentes móviles.....	34



2.4	Estándares de Agentes.....	36
2.4.1	MASIF .....	36
2.4.2	FIPA.....	42
2.5	Plataforma de agentes móviles para WSN.....	56
2.5.1	Aguilla.....	57
2.5.2	In-Motes .....	64
2.5.3	Arquitectura para WSN basada en FIPA.....	68
2.6	Conclusiones .....	72
3	Propuesta de hogar digital.....	74
3.1	Objetivos.....	74
3.2	Pasarela residencial.....	75
3.3	El libro blanco de Telefónica .....	78
3.3.1	El gestor del portal .....	81
3.3.2	Agregador de servicios .....	81
3.3.3	Integrador residencial.....	82
3.4	Pasarela Residencial basada en Agentes (PRA).....	82
3.4.1	Interfaces con las tecnologías dómicas instaladas en el HD .....	83
3.4.2	Plataforma de Agentes (Nodo Domótico).....	84
3.4.3	Plataforma de agentes para WSN .....	87
3.5	Tipos de agentes.....	90
3.5.1	Agente de primer nivel.....	91
3.5.2	Agentes de segundo nivel .....	92
3.5.3	Agentes tercer nivel .....	93
3.6	Servicios prestados en el HD .....	94
3.7	Software y Hardware.....	97
3.8	Funcionamiento de un servicio concreto .....	99
4	Conclusiones y trabajos futuros .....	103
	Bibliografía .....	105

## Índice de figuras

Figura 1 - Tareas en diagrama de Gantt .....	2
Figura 2. Nivel físico KNX.....	4
Figura 3. Componentes de un dispositivo KNX.....	5
Figura 4. Bloques de construcción definidos en DLNA .....	8
Figura 5. Categorías y clases de dispositivos DLNA.....	9
Figura 6. Arquitectura de un dispositivo uPnP .....	10
Figura 7. Tipos de dispositivos uPnP .....	11
Figura 8. Ejemplos de aplicaciones telemáticas .....	14
Figura 9. Pila de comunicaciones propuesta por WiMedia Alliance .....	14
Figura 10. Aplicación Cliente/Servidor .....	17
Figura 11. Ejemplo de funcionamiento de agentes.....	18
Figura 12. Características fundamentales de un agente .....	21
Figura 13. Tipos de agentes según movilidad.....	22
Figura 14. Ejemplo de movilidad de un agente .....	25
Figura 15. Término lugar.....	26
Figura 16. Viaje de un agente .....	27
Figura 17. Entrevista entre agentes.....	27
Figura 18. Sistemas multiagente .....	28
Figura 19. Ejemplo: Dos SA en una misma máquina .....	29
Figura 20. Ejemplo: Interconexión de los SA de dos máquinas.....	30
Figura 21. Ejecución asíncrona de tareas.....	32
Figura 22. Reducción de tráfico en la red .....	32
Figura 23. Robustos.....	33
Figura 24. Automatización del proceso de tareas distribuidas.....	33
Figura 25. Proceso de tareas local/descentralizado .....	34
Figura 26. Sistema de agentes MASIF .....	38
Figura 27. Servicios y utilidades CORBA.....	39
Figura 29. Implementaciones concretas de la Arquitectura Abstracta.....	46
Figura 30. Implementaciones de FIPA que utilizan otras implementaciones .....	47
Figura 31. Un agente FIPA se registra en un servicio de directorio .....	48
Figura 32. Estructura de un mensaje FIPA-ACL.....	49
Figura 33. Modelo de referencia de gestión de agentes FIPA.....	50
Figura 34. Ejemplo de AID de un agente FIPA.....	51
Figura 35. Estructura de un DF .....	52

Figura 36. Ciclo de vida de un agente FIPA.....	54
Figura 37. Modelo de agentes de Agilla .....	59
Figura 38. Direccionamiento en Agilla .....	60
Figura 39. Arquitectura de Agilla .....	61
Figura 40. Arquitectura de pila de un agente en Agilla .....	63
Figura 41. Arquitectura de In-Motes .....	65
Figura 42. In-Motes Bins.....	67
Figura 43. Arquitectura de agentes móviles para WSN basada en FIPA.....	69
Figura 44. Modelo de funcionamiento de la arquitectura.....	72
Figura 45. Adaptador Multimedia Interactivo.....	76
Figura 46. Marco de actuación de OSGi .....	77
Figura 47. Pasarela básica.....	78
Figura 49. Escenario de actuación de Telefónica .....	79
Figura 50. Integración de la pasarela en el escenario .....	80
Figura 51. Partes de la pasarela de Telefónica .....	81
Figura 52 - Escenario de actuación del Nodo Domótico .....	82
Figura 53 - Traducción de protocolos LonWorks-KNX .....	83
Figura 54 - Estructura genérica de conexión del nodo domótico .....	85
Figura 55 - Estructura específica de conexión del nodo domótico .....	85
Figura 56 - Ejemplo de traducción FIPA - KNX.....	87
Figura 57 - Registro de un agente .....	89
Figura 58 - Ejemplo completo del escenario de esta propuesta.....	90
Figura 59. Tipos de agentes.....	91
Figura 60. Árbol jerárquico de agentes.....	94
Figura 61 - Escenario hardware y software.....	98
Figura 62 - Agente Mayordomo.....	100
Figura 63 - Intercambio de mensajes entre agentes .....	100
Figura 64 - Agente climatización en PRA.....	101
Figura 65 - Agente detector trasladándose a WSN.....	101
Figura 66 - Orden KNX a aire acondicionado.....	102

## Índice de tablas

Tabla 1 - Tareas del TFM .....	2
--------------------------------	---

## 1 Introducción

La tecnología de agentes es una tecnología que ha presentado una gran actividad en los últimos años en el desarrollo de soluciones en el campo de las telecomunicaciones. Durante los últimos años está siendo estudiada por un gran número de universidades, investigadores y empresas de dicho ámbito. Esto es debido a las características y ventajas que presenta esta tecnología, principalmente la capacidad que tienen los agentes de adaptarse al entorno en el que se encuentren; y en el caso de los agentes móviles, a la capacidad que poseen para moverse dentro de una red. Por este motivo, y para comprender mejor esta tecnología, el tema central de este TFM (Trabajo Fin de Máster) gira en torno a los agentes móviles y sus aplicaciones.

En concreto, los objetivos que se han pretendido cubrir con este TFM han sido los siguientes:

- Conocer qué tecnologías inalámbricas se utilizan actualmente en los hogares domóticos y en qué consisten.
- Obtener una visión global del estado del arte actual de la tecnología de agentes, concretamente de los agentes móviles.
- Estudiar los diferentes estándares definidos para cubrir los objetivos planteados en la tecnología de agentes.
- Estudiar la aplicación de los agentes móviles en redes de sensores, y las plataformas existentes en dicho entorno, ya que, debido a sus características, los agentes móviles son muy aptos para trabajar en entornos con recursos limitados.
- Definir un modelo de agentes en el hogar digital, donde se aplicarán todos los conocimientos obtenidos sobre los agentes a largo de este TFM.
- Describir brevemente el funcionamiento lógico de un posible servicio desplegado en este modelo de agentes “implantado” en el hogar digital.

La mayor parte de este TFM recoge un estado del arte tanto de las diferentes tecnologías inalámbricas utilizadas actualmente en el hogar digital para ofrecer servicios como de la tecnología de agentes, que finalmente se pretende introducir en la propuesta realizada. Para ello, ha sido imprescindible conocer bien a fondo cuales son los pilares o fundamentos sobre los que se basa esta tecnología y qué estándares recogen estos fundamentos.

Una vez conseguido un nivel de conocimiento elevado sobre este tema de investigación, se ha procedido a desarrollar una propuesta que introducirá esta tecnología en el hogar digital para poder soportar los diferentes servicios ofrecidos.

La motivación sobre este tema, viene de un proyecto en el que estuve involucrado en Telefónica I+D. En dicho proyecto, se elaboraba un servicio para el hogar digital para que los usuarios pudiesen reproducir, pausar, detener, trasladar diferentes contenidos multimedia entre los diferentes "renderers" (televisores, pc's, radios, etc). A partir de ese momento, los servicios desarrollados e implantados en el hogar digital acapararon toda mi atención.

El plan de trabajo seguido se puede observar en la Tabla I, dónde se muestra las diferentes actividades realizadas y los meses en los que se han estado trabajando en ellas.

Tabla 1 - Tareas del TFM

Nombre de tarea	Duración	Comienzo	Fin
Proyecto servicio HD Telefónica I+D	86 días	lun 01/11/10	lun 28/02/11
Estudio tecnologías inalámbricas en el HD.	31 días	lun 17/01/11	lun 28/02/11
Estudio tecnología de agentes móviles	20 días	mar 01/03/11	lun 28/03/11
Estudio de estándares de agentes móviles	16 días	lun 28/03/11	lun 18/04/11
Estudio de estándares de agentes móviles para WSN	21 días	lun 18/04/11	lun 16/05/11
Propuesta Hogar Digital con agentes móviles	31 días	lun 16/05/11	lun 27/06/11

En la siguiente figura, se puede observar las tareas mencionadas anteriormente con respecto al tiempo en el que se han realizado.

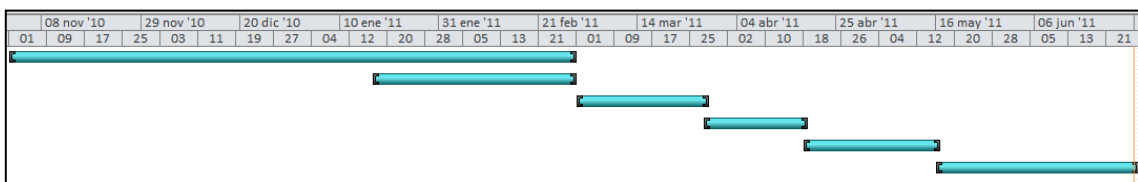


Figura 1 - Tareas en diagrama de Gantt

## 2 Estado del arte

Este apartado del estado del arte se ha dividido en dos partes muy diferenciadas. En primer lugar, se expondrá en qué estado se encuentran las diferentes tecnologías de red utilizadas y desplegadas en el Hogar Digital (HD), punto que contiene el primer subapartado de este estado del arte. Por otro lado, en el segundo bloque compuesto por los tres siguientes subapartados, se describirá en qué consiste la tecnología de agentes móviles, qué ventajas aporta, qué estandarizaciones se han creado para soportar sus características y despliegues de servicios, y qué estándares existe para las redes de sensores.

### 2.1 Tecnologías en el hogar digital

En la actualidad existe una gran multitud de tecnologías de red para la comunicación de dispositivos en el hogar digital, así como todos los servicios que se están desarrollando para su despliegue y disfrute de los usuarios. En este TFM (Trabajo Fin de Máster), se ha decidido incluir un total de seis tecnologías diferentes, pero con fines muy similares, que se exponen en los próximos apartados.

#### 2.1.1 KNX

La tecnología KNX [1] (hace unos años denominado KONNEX) surge con el objetivo de coordinar todos los dispositivos del hogar digital, y así poder obtener un perfecto funcionamiento de éste. Para ello, gran parte de la carga de esta tecnología, radica en su bus de control.

La tecnología KNX está aprobada como estándar internacional (ISO/IEC 14543-3 [2] [3]), estándar europeo (CENELEC EN 50090 y CEN EN 13321-1), estándar chino (GB/Z 20965) y estándar norteamericano (ANSI/ASHRAE 135). KNX, es el resultado de la unión de otras tres tecnologías, que son EIB (European Installation Bus), Batibus y EHS (European Home System); cada una de ellas aporta un nivel físico diferente, como se puede observar en la siguiente figura:

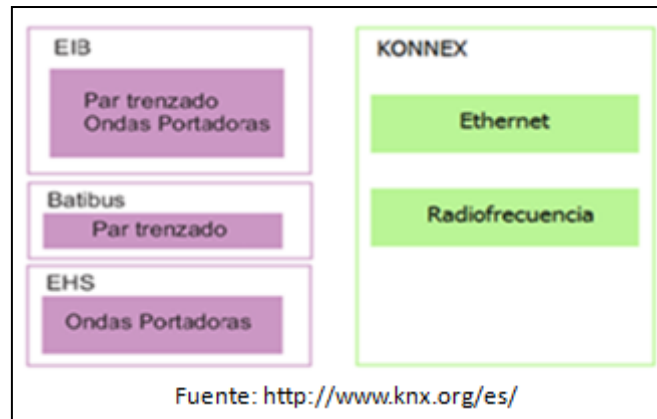


Figura 2. Nivel físico KNX

Entre los objetivos marcados por la tecnología KNX, destacan los siguientes puntos:

- El objetivo principal de KNX era conseguir crear un estándar único de comunicación entre los diferentes equipos. Esto fue debido a que LonWorks comenzó a consolidarse como una gran solución de comunicación de dispositivos. Por este motivo, EIB, Batibus y EHS se fusionan en KNX, ofreciendo un protocolo multimedia. Es decir, KNX permite transmitir señales por cable dedicado (BUS), Power Line, radiofrecuencia e incluso, por rayos infrarrojos a los diferentes dispositivos.
- Incorporar los buses domóticos en soluciones como la climatización u otros servicios cotidianos de los usuarios, para que de esta manera su impacto en el HD sea más profundo y tener más impacto la solución que ofrece.
- Mejorar las prestaciones de los medios físicos de comunicación, como por ejemplo en las soluciones de tecnologías de radiofrecuencias.
- Agregar al Hogar Digital (HD) aplicaciones Plug&Play que permita de esta manera una nueva forma de comunicación a través de conexiones USB.

Como se ha comentado en el primer punto de los objetivos marcados por KNX, este es uno de los pocos estándares abiertos a nivel internacional para utilizarlo en casas y edificios, lo que conlleva a ser uno de los pocos sistemas que se puede emplear para comunicar todos los dispositivos de una casa domótica; pero siempre y cuando, sólo se emplee un único lenguaje de programación para comunicar a todos los dispositivos (ETS - Engineering Tool Software). De este modo, se solucionan los problemas del uso de dispositivos aislados y de dispositivos de distintos fabricantes, eliminando así los posibles conflictos entre ellos.



Por otro lado, su modo de funcionamiento está dividido en tres categorías diferentes, que son las siguientes:

- *A-Mode*: proviene de “Automatic Mode” y consiste básicamente en que los dispositivos se configuran automáticamente, para que el usuario final no desarrolle ninguna función de instalación.
- *E-Mode*: proviene de “Easy Mode” y los dispositivos contienen una pequeña configuración inicial establecida de fábrica. Aún así, es necesario que los usuarios ultimen algunos detalles de la configuración final.
- *S-Mode*: proviene de “System Mode” y la configuración necesaria es la más complicada de las tres. Es por ello, que es necesaria la presencia de un técnico y/o especialista en KNX para que configure perfectamente la funcionalidad de los dispositivos instalados.

En cuanto a los dispositivos que KNX es capaz de controlar, se puede encontrar en iluminación, persianas y toldos, calefacción, aire acondicionado, sistemas de seguridad y vigilancia, audio y vídeo, etc. Para ello, estos dispositivos KNX están compuestos por los siguientes componentes:

- *Sensores*: dispositivo capaz de transformar magnitudes físicas o químicas (variables de instrumentación), en magnitudes.
- *Actuadores*: dispositivos que pueden provocar un efecto sobre un proceso automatizado.
- *Bus de datos*: elemento utilizado para el transporte de órdenes e información.
- *Adaptador del bus*: a través de este elemento se conectan los diferentes dispositivos de la instalación al bus.

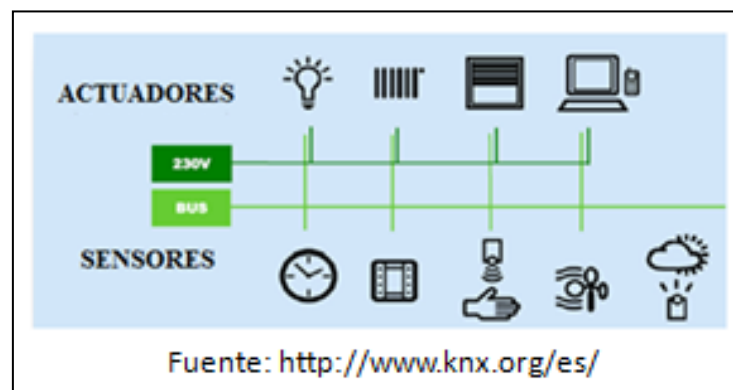


Figura 3. Componentes de un dispositivo KNX

### 2.1.2 LonWorks

LonWorks es una red de control creada por la empresa norteamericana Echelon [4]. Como el resto de redes de control, LonWorks está formada en un ámbito general por un conjunto de dispositivos denominados nodos, que se comunican entre sí para formar una aplicación de monitorización y/o de control. Cada uno de estos nodos está compuesto por uno o más sensores o actuadores.

El tipo de topología utilizada en los orígenes de las redes de control fueron soluciones centralizadas punto a punto, que consistían en un controlador “maestro” o “principal” que estaba cableado físicamente con el resto de componentes de la red, tales como sensores o actuadores (se les denominaba “esclavos”).

Aunque este tipo de topología centralizada aún se utiliza actualmente, son mucho más comunes las topologías distribuidas (más de un elemento de controlador maestro) que tienen por característica general ser más potentes, flexibles y ampliables con respecto a la primera opción. Es a partir de esta base, donde LonWorks comienza, y además se apoya en:

- Sistemas de control lo más similares posibles (estandarización), provocando una independencia con las aplicaciones desarrolladas.
- Debido a que potencia las soluciones distribuidas, esto provoca que las empresas consigan ahorrar un importe salarial considerable a largo plazo.

Estas características introducidas en LonWorks eran más que necesarias, ya que las redes de control no habían sido hasta el momento lo suficientemente eficaces como para tener un crecimiento acorde con lo que se les suponía. Por ello, los fabricantes han intentado buscar un paralelismo con la evolución de las redes de datos, que tanto han avanzado en los últimos años, llegando a la conclusión que lo más efectivo es sin lugar a duda la estandarización de una tecnología que predomine sobre las demás. Esto lo que supondría sería una gran adaptabilidad de los distintos productos y aplicaciones que salgan a la luz, ya que no necesitarían ningún tipo de adaptación para introducirlo en estas redes.

Debido a esto, LonWorks es una tecnología que tiene un futuro muy prometedor, ya que está respaldado por varias empresas del sector de redes de control como son las siguientes: Honeywell, Siemens Landis&Staefa, Trend Controls y Philips entre otros. Además de todo este apoyo, LonWorks tiene una accesibilidad muy amplia a las empresas que estén interesadas en introducirse en este sector, ya que posee herramientas de gestión, configuración y puesta en marcha muy sofisticadas para conseguir aplicaciones con cierta rapidez. Sin duda, las páginas dónde mejor nos podemos documentar sobre esta tecnología son en la página del fabricante, en la

página de la organización LonMark [5] y en la página de la feria de LonWorld Expo [6]. Los últimos acontecimientos, para que LonWorks se convierta en una tecnología predominante, es que se ha incluido en varios estándares como:

- Norma EIA-709.1. Especificación del Protocolo de Redes de Control.
- Incluido como parte de la norma de control de BACnet de la Sociedad Americana de Ingenieros de Calefacción, Refrigeración y Aire Acondicionado (ref. Norma: ANSI/ASHARE 135).
- Es el protocolo estándar para la Federación Internacional de Estaciones de Servicio.

Para finalizar, mencionar que en la actualidad LonWorks se considera normalmente como una posible alternativa a KNX.

### 2.1.3 ZigBee

ZigBee, también conocido como HomeRF Lite y diseñado por el grupo ZigBee Alliance [7], es una especificación de comunicaciones inalámbricas basado en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (WPAN - Wireless Personal Area Network). El propio grupo ZigBee Alliance lo define como el nuevo estándar de automatización del hogar digital.

Los objetivos de esta especificación son:

- Que las aplicaciones, que necesitan conexiones seguras con baja tasa de envío de datos, maximicen la vida útil de sus baterías.
- Permitir que las distintas aplicaciones domóticas, desarrolladas por diferentes fabricantes puedan comunicarse entre sí, garantizando seguridad y fiabilidad al usuario final.

Esta especificación proporciona un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización en radios digitales de bajo consumo.

Algunas de las características de ZigBee son: bajo coste, fácil implementación, transferencia de datos fiable a velocidades desde 20KB/s a 250 KB/s, rango de alcance de 10 a 75 metros y bajos consumo de energía.

Por lo tanto, el tipo de dispositivos que utilizará ZigBee en un hogar digital, serán dispositivos integrados en aplicaciones de baja transmisión de datos y bajo consumo de energía. Se pueden utilizar en aplicaciones de control industrial, recolección de datos,

aplicaciones de detección de humo o extraños en el hogar, etc. Debido al bajo consumo de energía, cada dispositivo podrá trabajar hasta 5 años sin cambiar sus baterías.

#### 2.1.4 DLNA

DLNA (Digital Living Network Alliance) consiste en un sistema el cual permite a los diferentes dispositivos que habitan en un HD (Hogar Digital), que se conecten entre sí para compartir contenidos, de forma fácil y sin necesidad de configuraciones complicadas. Este sistema DLNA, está apoyado por multitud de compañías inmersas en el mercado del HD, como son: Intel, HP, AMD, LG, Microsoft y Philips.

Uno de los objetivos futuros expuestos en esta propuesta DLNA, es que esta plataforma deberá interactuar con los distintos elementos inteligentes existentes en el HD. Por lo tanto, su funcionalidad será la de manejar y distribuir los distintos contenidos multimedia existentes en el HD entre los distintos dispositivos.

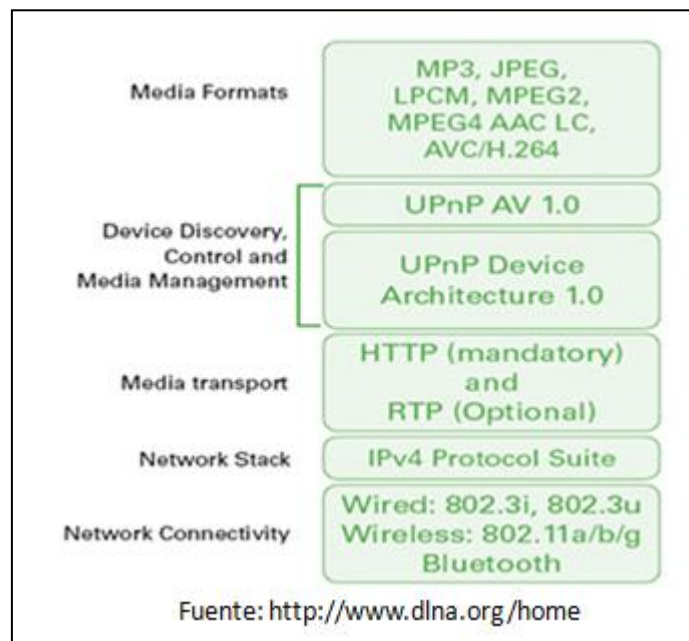


Figura 4. Bloques de construcción definidos en DLNA

Además, en esta propuesta DLNA se publica un conjunto de guías, dónde se especifican los distintos bloques de construcción disponibles para diseñar plataformas e infraestructuras software, como se puede observar en la figura anterior. Estos bloques de construcción, se agrupan en seis grupos y definen las siguientes funcionalidades:

- Conectividad transparente entre los dispositivos presentes en el HD.

- Procedimientos de descubrimiento, configuración de dispositivos y control unificados.
- Protocolos de streaming y formatos multimedia interoperables.
- Control y gestión de los contenidos y dispositivos interoperables.
- Mecanismos de calidad de servicio compatibles entre dispositivos.
- Mecanismos de autenticación y autorización de uso compatibles entre dispositivos.

Siguiendo con la guía de interoperabilidad definida en esta propuesta de DLNA, en ella se define un total de doce clases de dispositivos, agrupados a su vez en tres distintas categorías como se ilustra en la siguiente figura.

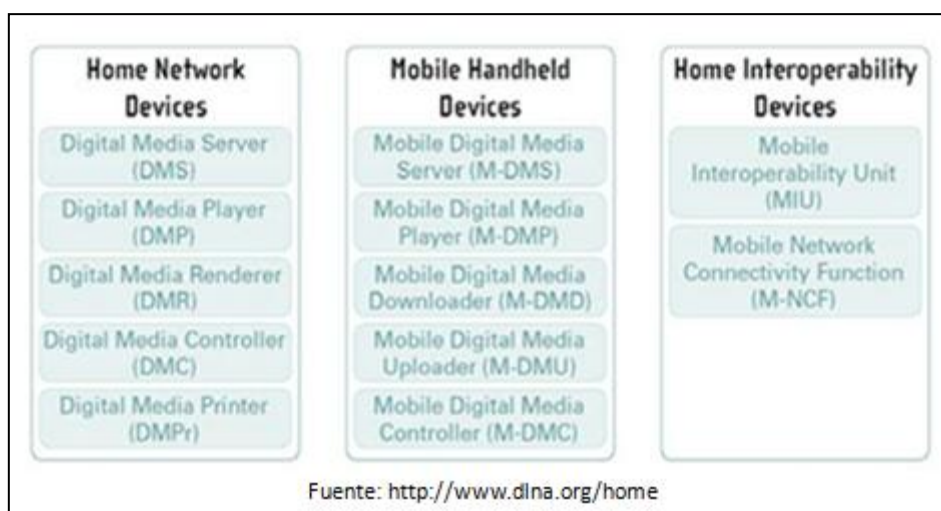


Figura 5. Categorías y clases de dispositivos DLNA

Dichas categorías tienen la siguiente funcionalidad dentro de esta plataforma DLNA:

- *Home Network Device*: Agrupa a cinco clases de dispositivos, que comparten los mismos tipos de funcionalidades en el HD con los mismos formatos multimedia e idénticos requisitos de conectividad en red.
- *Mobile Handheld Device*: Comparten los mismos tipos de uso que los Home Network Devices pero tienen diferentes requisitos en cuanto a conectividad y formatos multimedia.
- *Home Infrastructure Device*: Dispositivos pensados para que interactúen con los dos tipos de categorías de dispositivos anteriores.

Para finalizar con esta plataforma, indicar que el núcleo que habita en esta tecnología DLNA es uPnP (Universal Plug&Play).

### 2.1.5 uPnP

Tal como se ha mencionado, uPnP está en el centro de DLNA. Esta tecnología uPnP, impulsada por el uPnP Forum (uno de cuyos principales miembros es Microsoft), es una tecnología que proporciona básicamente servicios de direccionamiento, descubrimiento, descripción, control, detección de estado y descripción de las características de la presentación de los contenidos multimedia en el hogar digital.

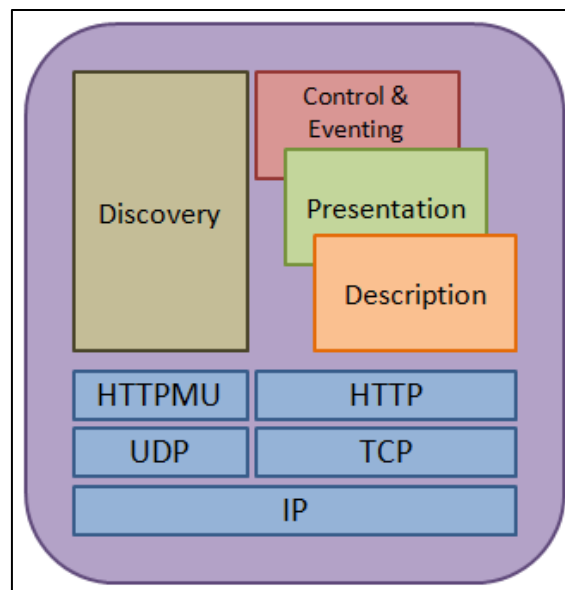


Figura 6. Arquitectura de un dispositivo uPnP

La figura anterior ilustra la arquitectura de un dispositivo uPnP. En ella se puede ver que uPnP ha sido creado para el mundo IP, sobre cuyos servicios UDP, TCP, HTTP se apoya. Dentro de las características más relevantes de uPnP es que no impone una plataforma de desarrollo en concreto, ni unos determinados formatos de objetos (bytecodes), ni métodos de distribución de ejecutables. uPnP se limita a mandar datos sobre la red y mantiene la privacidad sobre las implementaciones.

En uPnP se distinguen tres tipos de dispositivos:

- *User Control Point*: Dispositivo tal como una PDA o PC que permite el control de otros dispositivos uPnP que presentan un display adecuado.
- *Controlled Device*: Dispositivo uPnP que puede ser utilizado por otros dispositivos uPnP de la red.

- *Controlled Device Bridge*: Conecta la red uPnP a dispositivos no uPnP existentes en la casa.

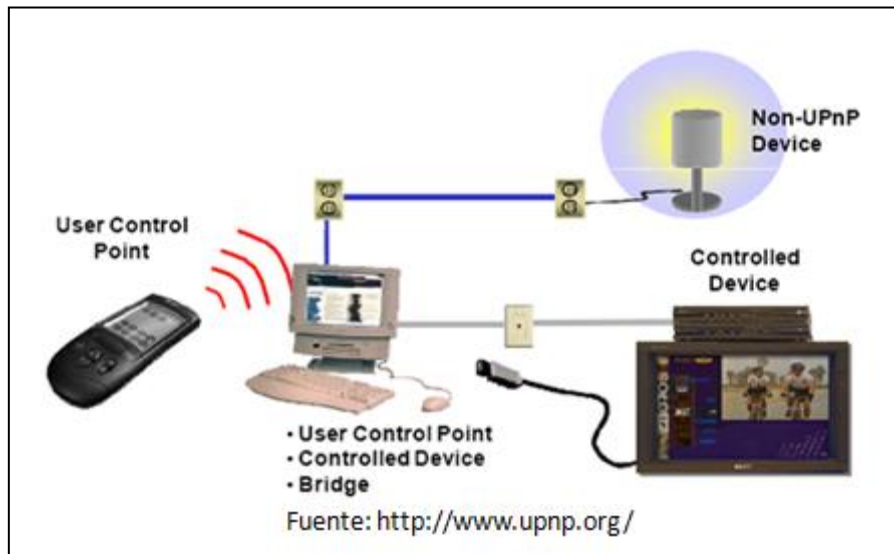


Figura 7. Tipos de dispositivos uPnP

En una conexión uPnP se han de recorrer los siguientes pasos:

a) *Direccionamiento de los dispositivos de red*: Cada dispositivo ha de adquirir un identificador que lo haga único en la red. Esta característica se apoya en el direccionamiento de la red que utiliza: IP.

b) *Descubrimiento*: Permite a un dispositivo uPnP anunciarse a la red utilizando el protocolo SSDP (Simple Services Discovery Protocol). Es posible buscar servicios de un determinado tipo.

c) *Descripción*: Cuando se ha encontrado un dispositivo del tipo que se necesita es necesario conocer su funcionalidad completa. Para ello se le reclama su descripción, lo que el dispositivo hará mediante un documento XML.

d) *Control*: Desde el Control Point pueden dar órdenes a dispositivos (p.ej. encenderse o apagarse) utilizando SOAP (Simple Object Access Protocol). Para ello el Control Point prepara un documento XML conteniendo las órdenes y posteriormente se formatea de acuerdo a SOAP para ser a continuación enviado al dispositivo mediante HTTP-TCP-IP.

e) *Eventos*: Una vez que un dispositivo o Control Point se ha registrado a un servicio de otro dispositivo, necesitan ser informados de los eventos que puedan ocurrir.

f) *Presentación*: El contenido multimedia se demanda mediante una petición HTTP. La manera en la que el dispositivo uPnP entrega la información está definida en la página de presentación que devuelve al cliente. Puede ser desde un plug-in a una página con comandos o cualquier otro elemento presente a partir de la V3.0 de HTML. uPnP Forum no normaliza más allá de este punto.

### 2.1.6 UWB

Ultra Wide-Band, es una tecnología de radiofrecuencia, que comenzó a desarrollarse en la década de los años 60 en entornos de las empresas relacionadas con instituciones militares y gubernamentales norteamericanas.

UWB normalmente describe una tecnología que se basa en la transmisión de impulsos de muy corta duración, con poca energía. Sin embargo, la definición de UWB propuesta por la FCC (Federal Communications Commission) [8] es “cualquier señal que ocupa un ancho de banda mayor de 500 MHz.”

Las transmisiones en UWB constan de una serie de impulsos de energía modulados. Para la modulación de la información de los distintos impulsos se varía su amplitud, polarización, fase o temporización. Debido a la corta duración de los impulsos, el espectro de frecuencias de una señal UWB es de un ancho de banda de unos 7 GHz, ya que se utilizan las frecuencias comprendidas entre 3.1 GHz y 10.6 GHz. Puesto que UWB debe distribuir la energía sobre un ancho de banda tan grande, la densidad espectral de dicha energía es muy pequeña, lo que se traduce en que no se producirán interferencias con otras señales que estén utilizando dicha porción del espectro radioeléctrico.

En UWB, el emisor y el receptor están activos durante periodos de tiempo muy cortos, esta característica da la posibilidad de que gran cantidad de dispositivos se encuentren en un mismo entorno, ya que cada uno de ellos tendrá una temporización y secuencia de impulsos diferentes, característica muy importante en las redes inalámbricas de área local (WLAN), lo que permite la existencia de gran número de transmisiones simultáneas. Para que tenga lugar la recepción es necesario que los receptores tengan información previa acerca de la temporización y secuencias de los impulsos de los emisores. Los amplificadores de entrada de los receptores están activados durante un periodo de tiempo corto, lo que le permite al receptor rechazar la mayoría de las señales no deseadas.

Según la FCC [8], UWB es una solución ideal para proporcionar conectividad inalámbrica dentro de un rango de entre 10 y 20 metros entre dispositivos electrónicos,



dispositivos móviles y periféricos de un ordenador, con una tasa de datos muy alta y un consumo de energía muy pequeño.

En cuanto a los servicios y aplicaciones que ofrece esta tecnología, decir que Ultra Wide-Band nació para ser utilizado en aplicaciones militares, pero, al ver las características de esta tecnología y su gran potencial, varias empresas e instituciones se dieron cuenta del amplio empleo que podía tener para otro tipo de aplicaciones comerciales, orientadas tanto al mercado de redes de área personal (WPAN) como de área local WLAN.

En el ámbito militar sus dos grandes aplicaciones fueron las comunicaciones y la vigilancia de espacios, como el suelo y subsuelo, ya que, los dispositivos UWB pueden ser utilizados para medir con gran precisión distancias y para capturar imágenes de alta resolución de objetos enterrados bajo tierra o detrás de superficies. Estas aplicaciones militares de implementación de radar y de localización, también pueden ser extendidas al mercado empresarial, principalmente en servicios de emergencia, vigilancia, sanidad o seguridad.

La tecnología UWB permite una transferencia muy alta de datos entre dos o más dispositivos que se encuentra a poca distancia. Es por ello que se han creado grandes expectativas en relación a su utilización para la fabricación de dispositivos móviles de bajo consumo, corto alcance y gran ancho de banda. Se apunta a un desarrollo más rápido del segmento orientado a la interconexión de dispositivos multimedia en un entorno residencial. Dentro de este ámbito, UWB se considera una tecnología muy útil para mejorar las prestaciones de los decodificadores de televisión, pudiendo distribuir la señal de vídeo y audio descodificada a todos los aparatos de televisión del hogar. De hecho, UWB ya ha empezado a ser utilizada en el desarrollo del denominado USB (Universal Serial Bus) inalámbrico.

En la actualidad nos encontramos con una gran cantidad de dispositivos que utilizan tecnologías inalámbricas para diferentes aplicaciones. Algunos ejemplos pueden ser la transferencia de datos de un ordenador a otro a través de Wi-Fi, la transmisión de música de un ordenador a un reproductor a través de Bluetooth, la sincronización de las agendas en una PDA, etc. Estos y otros ejemplos quedan recogidos en la siguiente figura.



Figura 8. Ejemplos de aplicaciones telemáticas

Por otro lado, puede observarse que la gran cantidad de servicios y de aplicaciones existentes en la actualidad utilizan distintas tecnologías para la comunicación y transferencia de datos. Existe un grupo de trabajo llamado WiMedia Alliance [9] que trabaja en las especificaciones de UWB para su uso en la electrónica de consumo, móviles y aplicaciones informáticas. WiMedia Alliance cuenta con el apoyo de más de doscientos miembros internacionales, corporaciones e instituciones de investigación para el desarrollo de especificaciones, tests de certificación y programas educativos. El esfuerzo de este grupo de trabajo se centra en utilizar Ultra Wide-Band para dar servicio a las aplicaciones a una alta velocidad, pudiendo ofrecer los mismos servicios que ofrecen otras tecnologías inalámbricas, como son Wi-Fi y Bluetooth, sin tener que realizar grandes cambios en las capas que se sitúan por encima del nivel físico, como puede observarse en la siguiente figura.

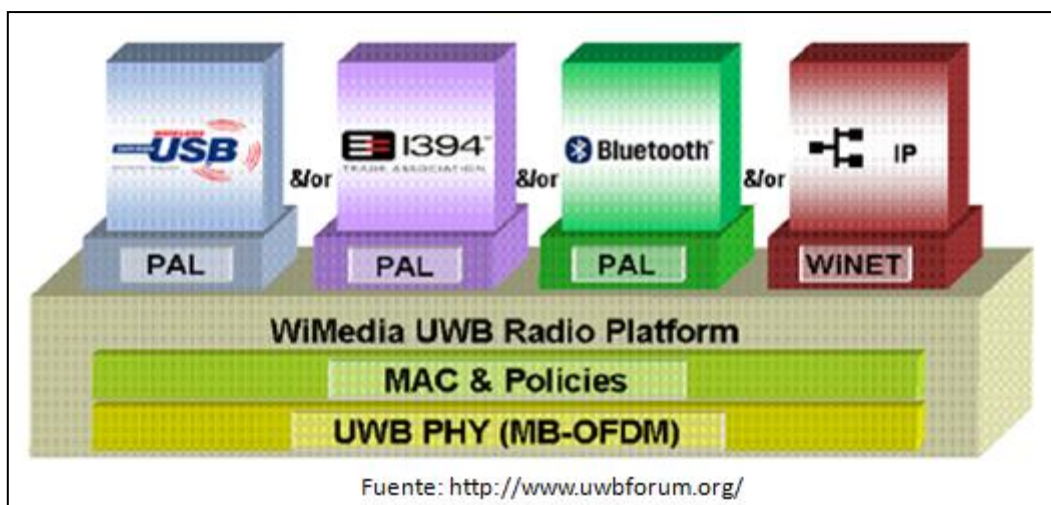


Figura 9. Pila de comunicaciones propuesta por WiMedia Alliance

De esta manera se realiza una convergencia de los servicios y aplicaciones ofrecidos por las distintas tecnologías inalámbricas, utilizando en el nivel físico UWB para la transmisión. Con este esquema se necesita un nivel PAL (Protocol Adaption Layer), para adaptar los distintos protocolos a las capas UWB.

### **2.1.7 Conclusiones sobre las distintas tecnologías en uso en el HD**

En estos primeros apartados se ha podido observar un conjunto de tecnologías que actualmente se están integrando en muchos de los HD. Sin embargo, entre ellas existe una gran diversidad en cuanto a sus funciones principales, soluciones, medios de transmisión, aplicaciones a las que están destinadas, etc.

Por hacer una mera clasificación fijándonos en sus características principales, podemos realizar un total de cuatro grupos. En el primero de ellos, se encontraría KNX y LonWorks, que aportan un bus de control de dispositivos (aunque también contienen soluciones inalámbricas). Por otro lado, tendríamos otro grupo que ocuparía ZigBee y que aporta comunicaciones inalámbricas centradas en los sensores y con unas transferencias de datos muy reducidas. El tercer grupo, formado por uPnP y DLNA (que es mucho más general e importante que uPnP, ya que DLNA incluye uPnP entre otras muchas cosas), van a ofrecer la posibilidad de transmitir contenidos multimedia a través de medios cableados. El último grupo, consistiría en una solución inalámbrica para transferencias elevadas de datos en unos periodos de tiempo muy reducidos y que estaría ocupado por UWB.

Entre todas ellas, ofrecen una amplia variedad de soluciones para soportar los diferentes servicios implantados en un HD. Debido a esta diversidad, no se puede denominar a unas soluciones mejores que otras, ya que cada una de ellas tiene características diferentes. Por tanto, en un mismo HD podrán convivir varias tecnologías diferentes, sin que ello signifique un problema, ya que a veces serán alternativas y en otras complementarias.

Lo ideal sería tener una única tecnología unificada que cubriese todas las necesidades en el HD pero, tanto por intereses comerciales como por la evolución de la tecnología, esto es muy difícil que pase. Por este motivo, se necesita soluciones que faciliten la integración de estas distintas tecnologías y permitan ofrecer a los usuarios un conjunto de servicios unificado, fácil de utilizar, y sin grandes complejidades de configuración y operación.

Con el objetivo de agrupar distintas tecnologías, este TFM se centra en introducir la tecnología de agentes en el HD, para que de esta manera, los usuarios vean los servicios de un HD de una manera unificada y no distribuida por un conjunto de tecnologías.

## 2.2 Tecnología de agentes

### 2.2.1 Introducción a los agentes

La tecnología de Agentes Móviles nace debido a la proliferación del uso de Internet y a la demanda de aplicaciones web más eficientes y autónomas. Inicialmente la mayoría de las aplicaciones en Internet estaban basadas en el paradigma Cliente/Servidor (por ejemplo RPC, RMI o CORBA), ya que permitían el acceso a la información existente en diferentes nodos de la red. Sin embargo, en ciertas aplicaciones han ido surgiendo ciertos paradigmas para poder ofrecer un funcionamiento diferente al modelo cliente/servidor. Uno de estos paradigmas son los agentes móviles.

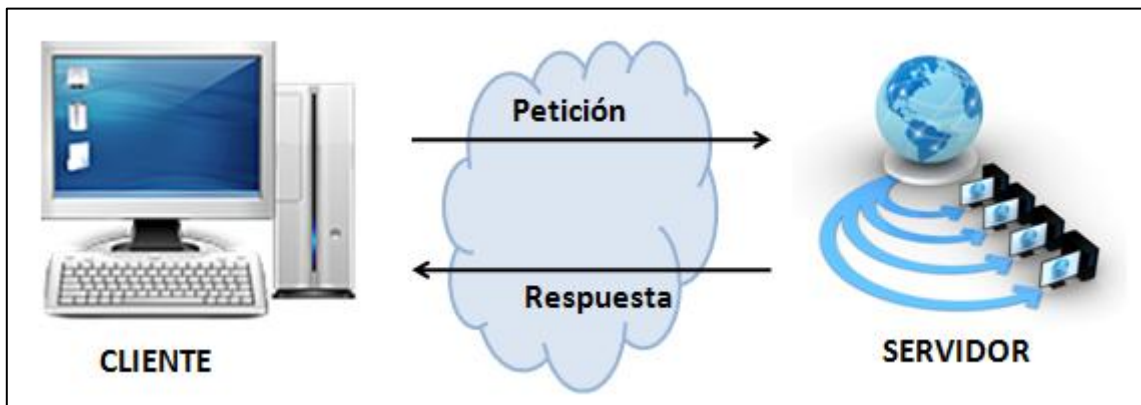


Figura 10. Aplicación Cliente/Servidor

Antes de la aparición de los Agentes Móviles, se probó con lenguajes multiplataforma (por ejemplo con Java Applets) que permiten el envío del código de las aplicaciones del servidor al cliente, disminuyendo el tráfico existente en la red, y además, se pueden ejecutar en diferentes sistemas operativos (Unix, Windows, Mac OS, etc). Con ello, se mejora el acceso a los datos ya que se encuentran en un entorno local. Pero a cambio tiene las desventajas de que sólo se puede descargar una vez, requiere de un plug-in en los navegadores, no puede ejecutarse si la JVM (Java Virtual Machine) no está ejecutándose y la descarga del código de las aplicaciones tiene que solicitarla manualmente el cliente en la página web dónde se encuentre [10].

Con los antecedentes de las aplicaciones Cliente/Servidor y los lenguajes multiplataforma, surgieron los Agentes Móviles para proveer aplicaciones autónomas y una manera diferente de generar aplicaciones que se adapten mejor a las:

- Redes de procesamiento de datos.

- Diferentes plataformas y arquitecturas.
- Redes de baja confidencialidad.

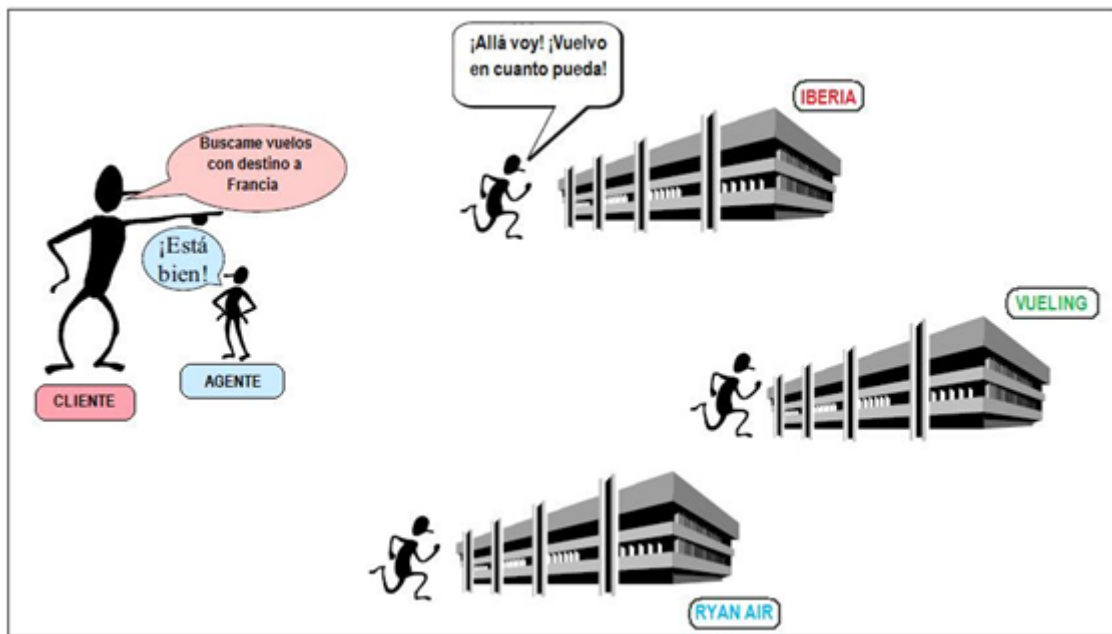


Figura 11. Ejemplo de funcionamiento de agentes

Sin embargo, este gran avance para mejorar la autonomía en el proceso de la información y de las aplicaciones, se ha visto frenado por distintos problemas, como puede ser la portabilidad o la seguridad. Aunque el problema de la portabilidad ha sido extinguido gracias a soluciones Java, la seguridad en la ejecución del código de otros servidores desconocidos aún no ha sido paliada y sigue siendo una grieta demasiado importante con la que se debe luchar, para que termine imponiéndose esta tecnología de Agentes Móviles.

### 2.2.2 Definición de agente

Dada la gran variedad de roles que un agente puede desempeñar, resulta bastante complicado dar una única definición exacta. Además, los investigadores utilizan distintas acepciones, lo que dificulta claramente que exista una definición ampliamente aceptada por la comunidad. Por ello, se aporta las siguientes definiciones:

- Ámbito general:* "Un agente es una pieza de software que realiza una tarea utilizando la información obtenida de su entorno para actuar de manera adecuada para completar la tarea exitosamente. El software debe ser capaz de adaptarse a sí mismo en base a los cambios que se producen en su

entorno, de modo que un cambio en circunstancias le permita aún obtener el resultado deseado” realizada por G.W. Lecky-Thompson [11].

- b) *Agentes inteligentes*: “Un agente inteligente es una entidad software que, basándose en su propio conocimiento, realiza un conjunto de operaciones para satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de éstos se lo requiere” [12].
- c) *Agentes móviles*: “Un agente móvil es un objeto especial que tiene un estado de datos (otros objetos no agentes, estructuras y bases de datos), un estado de código (las clases del agente y otras referencias a objetos) y un estado de ejecución (el control de procesos que se ejecutan en el agente)” dada por Fritz Hohl [13].
- d) *Según el Object Management Group*: “Un agente es un programa de ordenador que actúa autónomamente en nombre de una persona u organización” [14].
- e) *Según I.B.M. Aglets*: “Los agentes inteligentes son entidades programadas que llevan a cabo una serie de operaciones en nombre de un usuario o de otro programa, con algún grado de independencia o autonomía, empleando algún conocimiento o representación de los objetivos o deseos del usuario” [15].

Como se ha podido observar, ninguna de estas definiciones de Agente es unánime y completa, a la vez, tampoco es compartida por todos los desarrolladores e investigadores. Por lo que, para obtener una imagen general sobre lo que significa y aporta esta tecnología, nos centraremos en examinar detenidamente las características que se espera de un agente.

### 2.2.3 Características de un agente

Las características más representativas de un agente son las que se exponen a continuación [16]:

- a) *Autonomía*: un agente es autónomo o semiautónomo, por lo que opera sin la intervención directa de los humanos, decidiendo aspectos como, para agentes móviles, dónde, cuándo y cómo debe migrar. Esto requiere que el agente tenga un mínimo control sobre sus acciones y de su estado interno.
- b) *Orientado hacia un objetivo final*: los agentes están destinados a ejecutar una tarea determinada. Para ello, deben decidir la mejor manera para obtener el objetivo final.
- c) *Reactividad*: la inteligencia que poseen los agentes les permite percibir el entorno en el que se encuentran y actuar en consecuencia según los cambios que se produzcan. Por lo que, pueden estar durante largos periodos de

tiempo en estado pasivo y despertar para reaccionar a cambios producidos en su entorno, modificar su conducta por acciones realizadas por otros agentes o por la intervención del usuario.

- d) *Proactividad*: los agentes también pueden operar con una iniciativa propia para obtener el objetivo final por el que fueron creados y no sólo basándose en cambios producidos en su entorno. Esto significa, por ejemplo, que los agentes pueden suspender su ejecución, trasladarse a otro host distinto del que se encontraban y reanudar su ejecución en el punto donde se suspendió. Todo esto, sin la necesidad de existir un cambio en su entorno.
- e) *Sociable*: los agentes tienen la capacidad de comunicarse con los usuarios o programadores, así como con otros agentes e incluso con el medio en el que se encuentra ejecutándose.
- f) *Movilidad*: es la capacidad que muchos agentes poseen y deben ser capaces de suspender su ejecución, para que posteriormente la reanuden en un host diferente, sin que ello interfiera en la correcta ejecución del agente.
- g) *Continuidad temporal*: un agente debe poder ejecutar sus procesos ya esté en un estado activo o pasivo (cuando un agente no ejecuta ninguna de sus acciones programadas).
- h) *Duplicidad*: permite que los agentes puedan clonarse (la ejecución de cada uno de los clones creados se ejecutan en hilos diferentes para no interferir entre ellos) para dividir sus tareas y así, poder obtener el objetivo final de una manera más sencilla, eficaz y rápida.
- i) *Aprendizaje*: los agentes son capaces de “memorizar” y automatizar sus tareas por medio del aprendizaje. De esta manera, son capaces de aumentar su rendimiento y realizar las tareas de una manera natural.

Hasta aquí llegan todas las características generales de los agentes; aunque algunos investigadores centrados en la IA (Inteligencia Artificial) consideran que el significado de agente es mucho más fuerte y específico, aportando otras características suplementarias como benevolencia, racionalidad, adaptabilidad y colaboración.

Como se puede observar, la controversia sobre lo que un agente puede ser y aportar, no sólo llega a una definición estándar sobre este asunto, sino que también aparece a la hora de especificar las características propias de esta tecnología. También es importante detallar que los agentes actuales aún no poseen todas las características defendidas por los investigadores centrados en la IA, y de hecho tampoco es necesario que las tengan para ser útiles en muchas aplicaciones; pero según se avanza en este campo, sí que van existiendo sistemas de agentes que poseen, al menos en cierta medida, gran cantidad de ellas.



## 2.2.4 Clasificación de agentes

En cuanto a la clasificación de los agentes, todo vuelve a depender del punto de vista de los investigadores, ya que existen diferentes maneras de encuadrarlos dentro de un tipo de clasificación u otra. Los diferentes tipos de clasificación se basan en las características propias de los agentes, por la tecnología que los soporta, por la función que desarrollan o por una mezcla de todas ellas. Aunque sí es verdad, que existe una clasificación predominante sobre las demás y ésta se basa en las características propias de los agentes.

### a) Los agentes en cuanto a su característica fundamental

Esta clasificación se basa en las características más importantes o con mayor peso de todas las que se han detallado en el apartado anterior: autonomía, movilidad e inteligencia. A partir de ellas se diferencian los siguientes tipos de agentes:

- *Agentes Autónomos*: se trata de agentes que se ejecutan en un hilo propio y que poseen la cualidad de elegir decisiones importantes sin la intervención directa del usuario o de otro programa.
- *Agentes Móviles*: son capaces de suspender su ejecución, almacenando su estado para poder trasladarse de un servidor a otro, y una vez realizado su viaje, recuperar su proceso de ejecución en el mismo momento dónde se interrumpió.
- *Agentes Inteligentes*: tienen una capacidad de razonamiento que les permite obtener la mejor opción de las disponibles en un momento determinado (situación del ambiente en el que se encuentran). Además, son capaces de decir qué acción llevar a cabo en base al estudio del ambiente, aprender del entorno y hasta de realizar tareas complicadas.

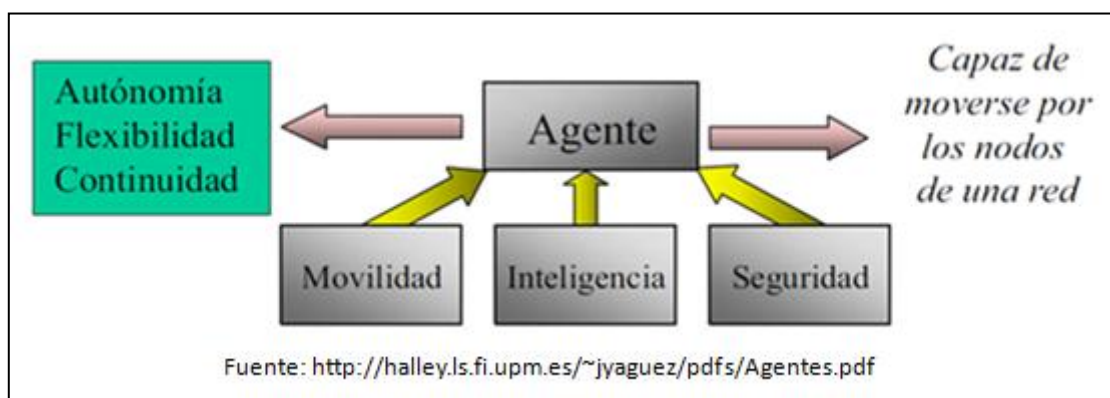


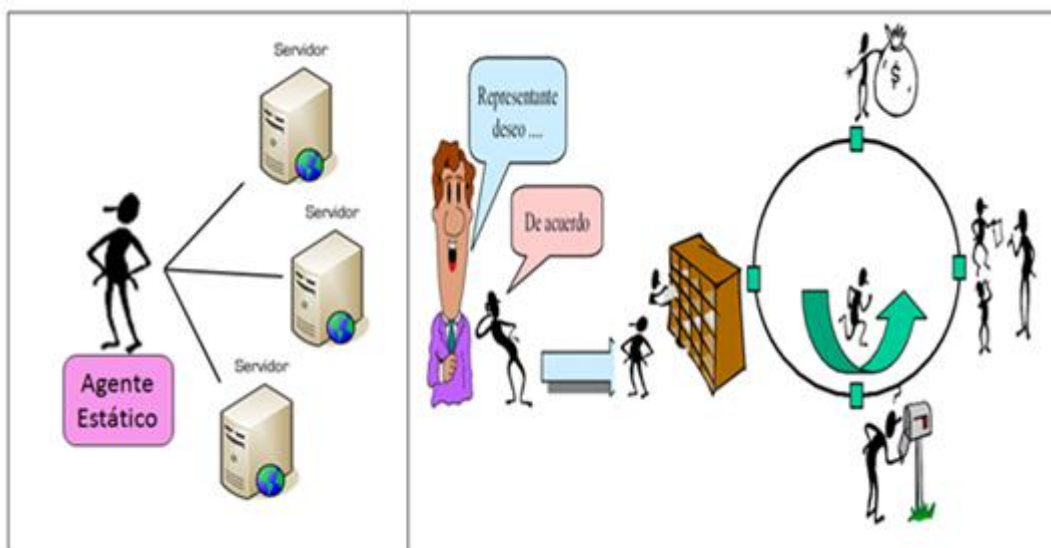
Figura 12. Características fundamentales de un agente

A través de la figura anterior, se puede observar las características más importantes que un agente puede poseer, este es capaz de aportar autonomía, flexibilidad, continuidad y la capacidad de moverse de un nodo a otro dentro de la red.

### **b) Los agentes en base a su movilidad**

Los agentes tienen la posibilidad de ser estáticos, es decir, se ejecutan dentro de la máquina donde fueron lanzados inicialmente, o ser móviles. En este último caso, los agentes pueden viajar de un servidor a otro obteniendo los datos o funcionalidades que requieren para obtener su objetivo.

- *Agentes Estacionarios o Estáticos:* estos agentes se ejecutan dentro de un único sistema informático en el cual fueron lanzados. Para realizar sus tareas u obtener el resultado esperado de ellos, se comunican con el mundo exterior a través del intercambio de mensajes.
- *Agentes Móviles:* tienen la capacidad de poder trasladarse de una máquina a otra para obtener la información o recursos necesarios para llevar a cabo sus tareas. Una vez conseguido su objetivo por el cual fue creado, vuelve a su máquina de origen.



Fuente: <http://halley.ls.fi.upm.es/~jyaguez/pdfs/Agentes.pdf>

Figura 13. Tipos de agentes según movilidad

### **c) Diversos tipos de agentes**

En esta última clasificación de los agentes, se realiza una nueva clasificación que intenta centrarse ante todo en la funcionalidad, para dar lugar a nuevos tipos de

agentes. Es importante indicar que dicha clasificación, que a continuación se expone, es la adoptada por el investigador Hyacinth Nwana [17].

- *Agentes de Interfaz:* son destinados a aportar una interfaz gráfica al usuario para ayudarlo a realizar tareas determinadas. Esta interfaz gráfica, le permite al usuario olvidarse de cierta manera de los procesos que van realizando los diferentes agentes que tiene activos y centrarse únicamente en los valores que van devolviendo tras ejecutarse. Lo que finalmente ocasiona que los agentes tengan un grado alto de autonomía con respecto de los usuarios y que el agente de interfaz termine con el tiempo aprendiendo tanto del usuario como de los agentes que se ejecutan dentro de su entorno.

- *Agentes Colaborativos:* su funcionalidad es interactuar y cooperar con otros agentes para conseguir un fin determinado. Para un mejor entendimiento, se aplica la idea de “divide y vencerás”, ya que para llevar a cabo un problema complejo es mejor dividirlos en pequeños módulos que se ayudan entre sí.

- *Agentes de Información:* estos agentes también conocidos como Agentes de Internet surgen debido al crecimiento constante de información disponible en la red. Por ello, la necesidad de herramientas que seleccionen, manipulen y recopilen la información deseada en un instante determinado.

- *Agentes Reactivos:* también denominados como Agentes Situacionales y se destinan a aplicaciones de ambiente real. Por ello, su tiempo de respuesta debe de ser mínimo y la idea de cómo y dónde son aplicados, dependen de la situación donde se toma la decisión de llevar a cabo una acción específica.

- *Agentes Híbridos:* surgen de la idea de combinar dos o más tipos de agentes para obtener un agente con una funcionalidad muy específica y completa en una aplicación determinada.

- *Agentes Inteligentes y Móviles:* estos dos últimos tipos de agentes son los mismos que se han detallado en los apartados anteriores.

#### **d) Evolución de los agentes**

Como se ha podido observar, existe una gran variedad de tipos de agentes, lo que ha provocado que su evolución haya sido muy variada.

La primera empresa que invirtió capital sobre la tecnología de agentes fue General Magic, que junto con las empresas Motorola y AT&T, consiguieron desarrollar una tecnología, llamada Telescrip, para dispositivos de computación portable.

La siguiente evolución llegó cuando Telescrip fue adaptado para trabajar con servidores web, dando lugar a la tecnología Tabriz AgentWare. Esta nueva tecnología, tenía como objetivo gestionar y ejecutar aplicaciones basadas en agentes, las cuales se ejecutaban dentro de los servidores.

Con el paso del tiempo, el sector de IA comenzó a centrarse en la tecnología de agentes, debido a su naturaleza inteligente. Su objetivo era el de conseguir que esta tecnología de agentes, alcanzase un nivel de inteligencia similar a la de un humano. Pero además de este objetivo extremadamente ambicioso y a muy largo plazo, la tecnología de agentes proporciona ya ventajas para el desarrollo de aplicaciones con determinadas características, como las que se dan en la interacción con Internet.

El gran crecimiento de la red Internet provocó la necesidad de crear nuevas herramientas o tecnologías, para poder satisfacer a todos los usuarios. Esta nueva situación, se puede ver reflejada en la aparición de un gran número de entornos de trabajo basados en los agentes móviles que realizaran tareas en Internet para sus usuarios humanos. A continuación, se muestra algunos de estos nuevos entornos de trabajo:

- Voyager (ObjectSpace) [18].
- Aglets Workbench (IBM) [19].
- Grasshopper (IKV++).
- SOMA "Secure and Open Mobile Agent" (MOSAICO project) [20].
- Agent Tcl (Dartmouth College).
- ARA "Agents for Remote Action" (University of Kaiserslautern) [21].
- Concordia (Mitsubishi Electric Information Technology Center America) [22].
- Java Agent Template (H. Robert Frost) [23].
- JATLite (Stanford University) [24].

## 2.3 Agentes móviles

Los agentes móviles, como bien indica su nombre, se diferencian de los demás tipos de agentes debido a que tienen la capacidad de moverse del lugar dónde fueron creados (normalmente denominados sistemas multiagente). Por lo tanto, se puede decir que un agente móvil tiene la capacidad de moverse de una máquina a otra a través de la red.

Esta característica de poder trasladarse de una máquina a otra, permite al agente interactuar directamente, de manera local, con otros objetos, para poder obtener la funcionalidad o información que le es necesaria para finalizar su tarea.

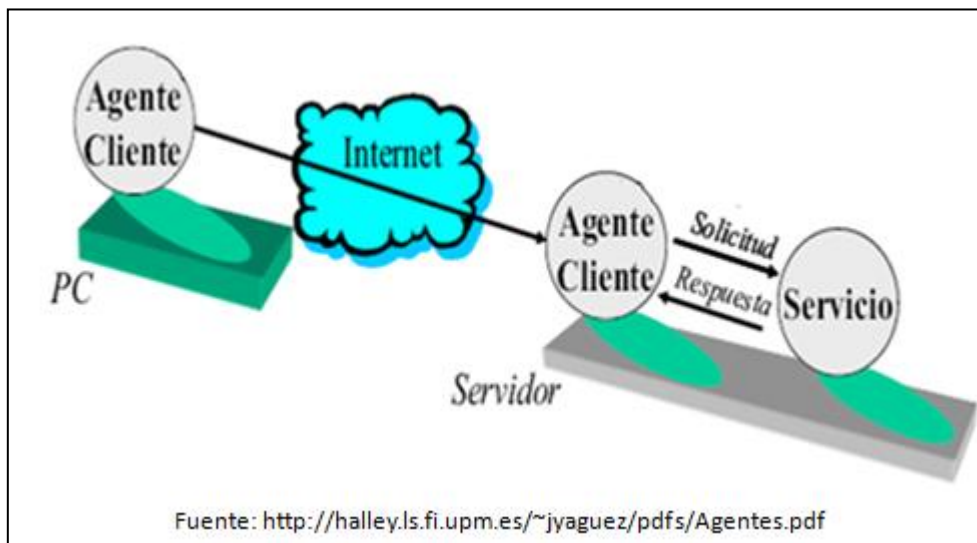


Figura 14. Ejemplo de movilidad de un agente

Sin embargo, dicha movilidad que este tipo de agente tiene por naturaleza, no sería posible sin los procesos de serialización y deserialización. Estos procesos permiten representar el estado de un agente en una secuencia de bytes, que posteriormente se enviará al destino por medio de la red. En este punto, se realizará el proceso inverso, de la serie de bytes se obtendrá el estado del agente (deserialización).

A continuación, se analizarán algunos detalles importantes que los agentes móviles presentan (conceptos básicos y tecnología) y algunas de las ventajas y posibles aplicaciones.

### 2.3.1 Conceptos de agentes móviles

Como anteriormente se comentó, General Magic fue la primera compañía que comenzó a desarrollar esta tecnología de agentes y la que estableció los primeros

conceptos sobre la misma. Dichos conceptos fueron tan bien definidos y con tanto acierto, que actualmente muchos de los nuevos diseños que han salido a la luz los han mantenido.

Este conjunto de conceptos se definieron con el fin de obtener una estructura general, que debe mantener la tecnología de agentes para un funcionamiento correcto. Los conceptos más relevantes que se describieron fueron los siguientes [16]:

- a) *Agente*: En apartados anteriores se vieron varias clasificaciones de los diferentes agentes que existen. Pues bien, como no era de extrañar, en este caso, los agentes que nos van a interesar son los agentes móviles. Estos se diferencian de los demás, por su capacidad de moverse por distintos “lugares” y así poder interactuar con el objeto deseado localmente.
- b) *Lugar*: Un lugar es el sitio donde los diferentes agentes viajan y se ejecutan para conseguir finalizar sus tareas. Un lugar es un sitio restringido, ya que los agentes pueden entrar o no según el nivel de seguridad que esté establecido. Además en él, los diferentes agentes tienen la posibilidad de interactuar entre ellos de manera local.

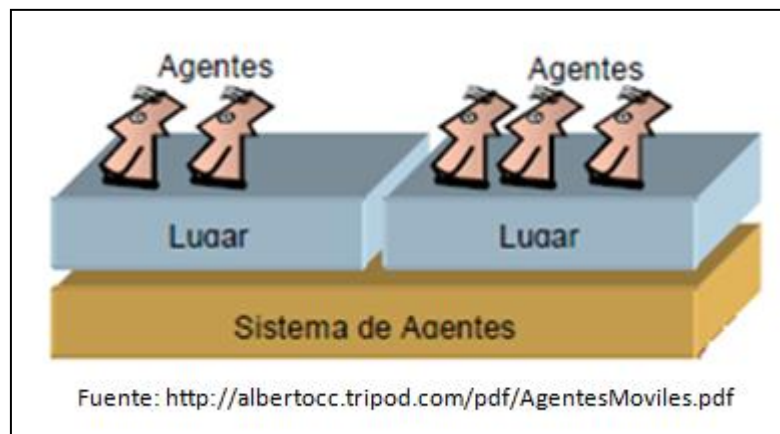


Figura 15. Termino lugar

- c) *Viajes*: Es el hecho que les diferencia de los demás tipos de agentes y de los distintos sistemas de programación remota. Un viaje consiste en que un agente abandone su sistema multiagente origen para trasladarse a otro sistema multiagente destino.

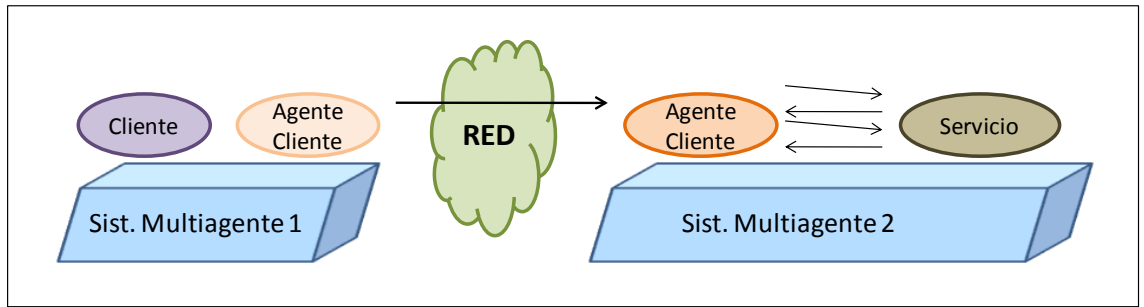


Figura 16. Viaje de un agente

- d) *Entrevistas*: Este término o concepto entrevista se refiere a las llamadas a métodos que se realizan entre los diferentes agentes que residen en un mismo lugar. Dichas entrevistas, únicamente se pueden realizar entre agentes que se encuentren en un mismo lugar, como se ha dicho anteriormente, lo que motiva a que los agentes viajen a diferentes lugares y sistemas multiagente.

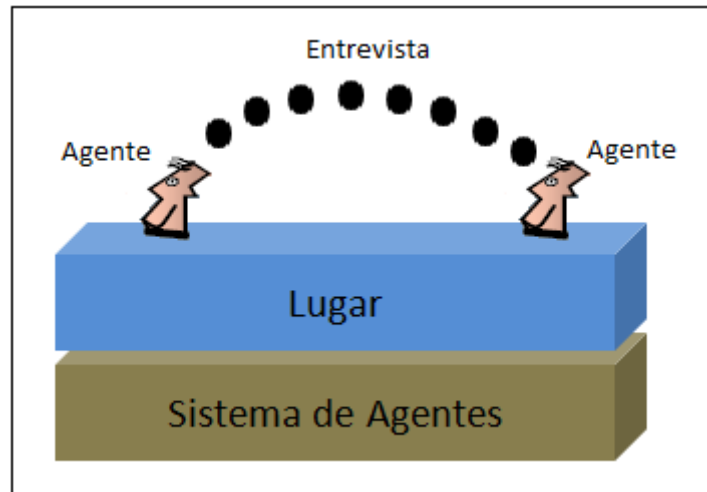


Figura 17. Entrevista entre agentes

- e) *Conexiones*: Este concepto de conexión se creó fundamentalmente para el intercambio de información entre agentes que se encontraban en distintos lugares.
- f) *Autoridades*: Todos los agentes y lugares contienen este término, ya que indica a qué persona u organización pertenecen.
- g) *Permisos*: Las autoridades definen qué capacidades les son otorgadas a los distintos agentes y lugares. Estos permisos, por tanto, definen qué pueden hacer o no.

### 2.3.2 Tecnología de los agentes móviles

La tecnología de agentes móviles proporciona un gran potencial, ya que como ya ha ido comentando anteriormente, los diferentes agentes tienen la posibilidad de interactuar entre ellos, para obtener sus propios objetivos. Pero hasta ahora, no se explicado nada sobre los entornos o plataformas donde conviven los diferentes agente: los sistemas multiagente.

De forma breve, podemos observar en la figura que a continuación se muestra, qué estructura presenta los agentes móviles en estos sistemas multiagente.

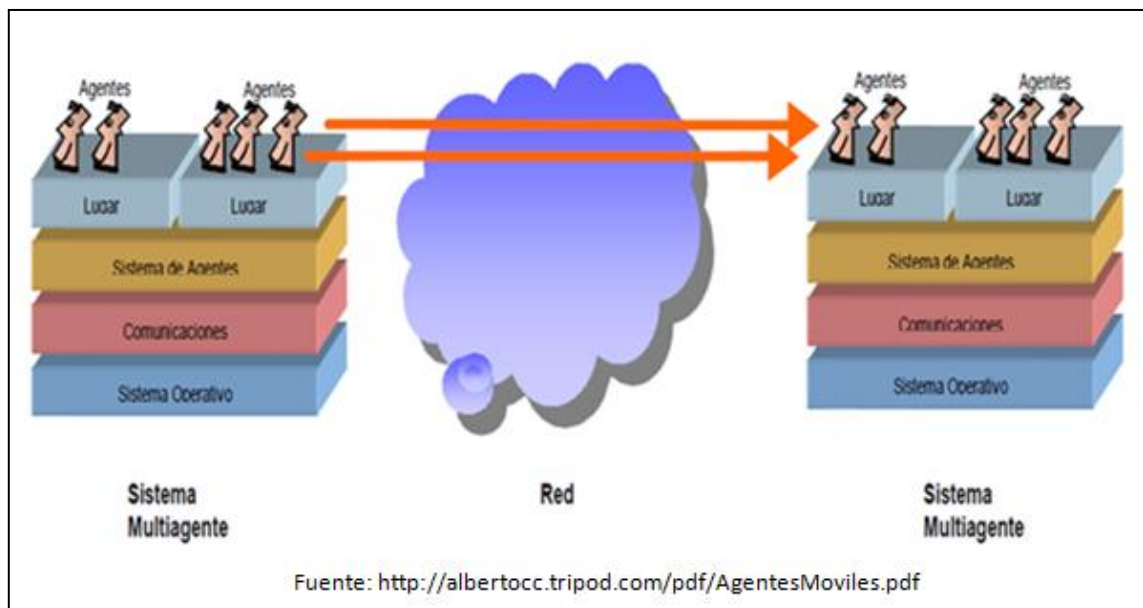


Figura 18. Sistemas multiagente

Por otro lado, este subapartado también se centra sobre el lenguaje de programación de agentes y el lenguaje de comunicación entre los agentes dentro de un mismo sistema de agentes.

a) *Sistema multiagente:*

Un sistema multiagente, como anteriormente se ha comentado, es una plataforma o entorno de ejecución de agentes. A un sistema multiagente, se le va a exigir fundamentalmente la creación, ejecución inicial, suspensión y finalización de agentes. Por tanto, va a ser responsable de las siguientes funciones [25]:

- Coordinar las acciones entre los diferentes agentes.
- Garantizar la seguridad del sistema para evitar la llegada de agentes maliciosos.



- Permitir la asociación con una autoridad para que identifique a una persona u organización.
- Describir de alguna manera parte del perfil de un agente. Dicho perfil queda definido por los tres siguientes parámetros: Tipo del SA (Sistema de Agentes), lenguaje y método de serialización.
- Disponer de uno o varios lugares (entornos específicos de ejecución) en función de los objetivos de los agentes. Como se puede observar en las siguientes figuras, un lugar puede albergar uno o más agentes y, a su vez, en una misma máquina puede existir uno o más SA. Si un lugar dispone de más de un agente, cada uno de ellos ofrecerá una clase de servicio diferente para ese mismo lugar.

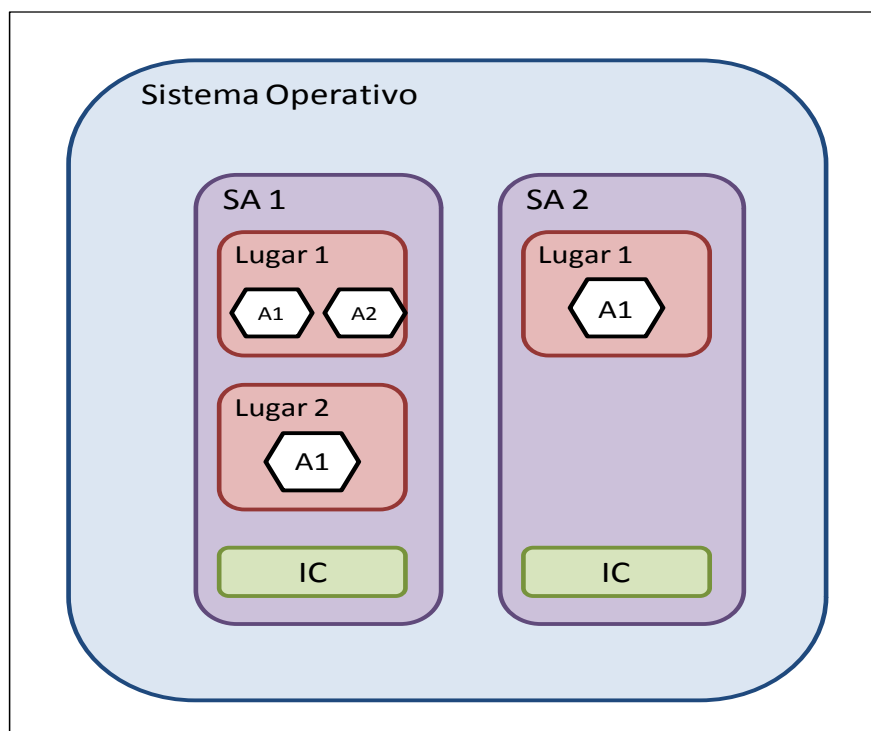


Figura 19. Ejemplo: Dos SA en una misma máquina

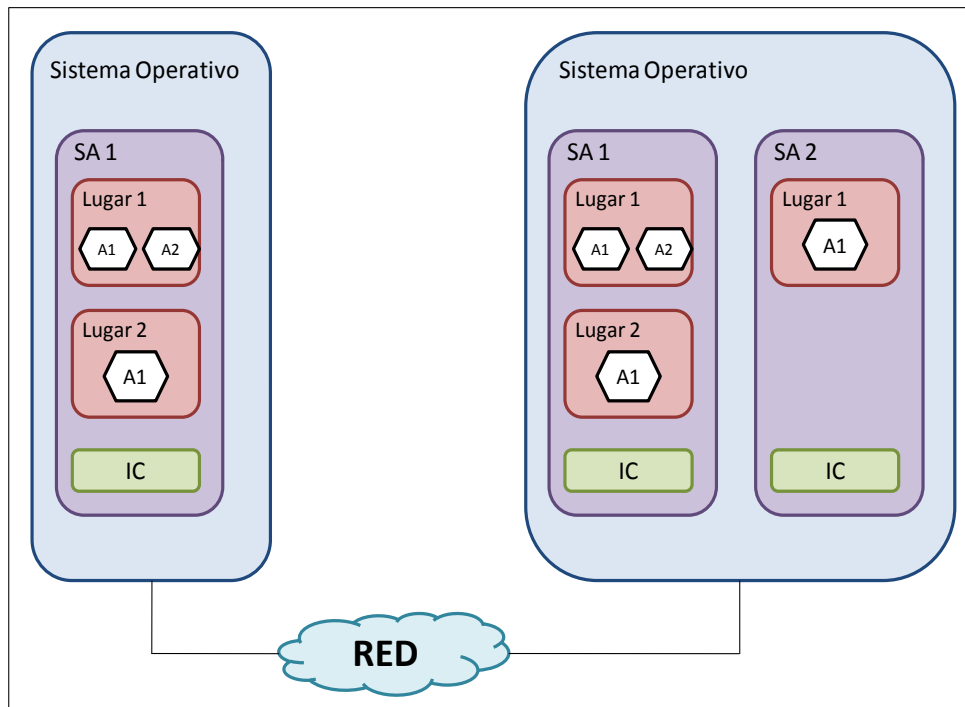


Figura 20. Ejemplo: Interconexión de los SA de dos máquinas

- “Disponer de un servicio de nombres que permita encontrar agentes y lugares locales. En este contexto, cada agente dispone de un identificador global y único formado por la concatenación de las siguientes informaciones: Autoridad representada, SA, lugar, un valor o identificador y la dirección IP de la máquina.”
- “Disponer de una infraestructura de comunicaciones (IC) que ofrezca los pertinentes servicios de transporte entre los SA.”

b) *Lenguajes de codificación de agentes y comunicación entre agentes*

La tecnología de agentes móviles, no está limitada a un grupo cerrado de lenguajes de programación, y por lo tanto se puede utilizar cualquier lenguaje. El único inconveniente de elegir un lenguaje de programación que no está específicamente diseñado para esta tecnología de agentes, es que el programador deberá crear o desarrollar al agente móvil desde el principio, teniendo que agregar todas las funcionalidades y características que dicho agente lleva consigo.

El lenguaje de programación ideal, sería aquel que proporcione las características de ser un lenguaje orientado a objetos, interpretado y que sea capaz de soportar las funcionalidades y características de los agentes móviles.

Algo similar sucede con los lenguajes para crear sistemas de comunicación entre agentes, ya que cualquier lenguaje de programación sirve. En este caso, dicho lenguaje

elegido por el programador deberá soportar las funcionalidades de creación, recepción y envío de datos correspondiente a los distintos mensajes necesarios para una correcta comunicación entre dos agentes.

Dentro de los grupos de lenguajes, los cuales cumplen las características anteriormente descritas, los más destacados son los siguientes:

- JAVA [26].
- C++ [27].
- Tcl (Tool Command Language) [28].
- Telescript.

### 2.3.3 Ventajas de los agentes móviles

Como se ha podido comprobar durante este apartado, la tecnología de agentes tiene un futuro prometedor ya que posee más de una ventaja importante, para finalmente poder competir con cualquier otra tecnología en este mundo de los sistemas distribuidos. Entre las ventajas que incorporan los agentes móviles, sobresalen por encima de las demás, poder desarrollar aplicaciones autónomas, la reducción de tráfico en la red y la facilidad que ofrecen de ejecutar tareas en paralelo y así poder ser más eficaces y rápidos.

Sin embargo, la tecnología de agentes no sólo aporta tres ventajas, ya que se pueden observar muchos beneficios, como los que a continuación se exponen:

- a) *Eficiencia:* La tecnología de agentes tiene una eficiencia muy superior que cualquier otra tecnología distribuida, ya que provoca que la red lleve menos mensajes, debido a que los agentes se mueven de un lugar a otro para trabajar localmente en los diferentes servidores
- b) *Ejecución asíncrona de tareas:* Otro beneficio que se obtiene al utilizar esta tecnología, es que desaparecen las conexiones permanentes a la red para que una aplicación se ejecute correctamente. Por lo tanto, un cliente puede crear un agente, para que este viaje al servidor y allí, ejecute las tareas programadas, teniendo éste, el control de ejecución de la tarea. De esta manera, al cliente se le da la opción de terminar o continuar haciendo otras cosas, hasta que el agente vuelva de nuevo.

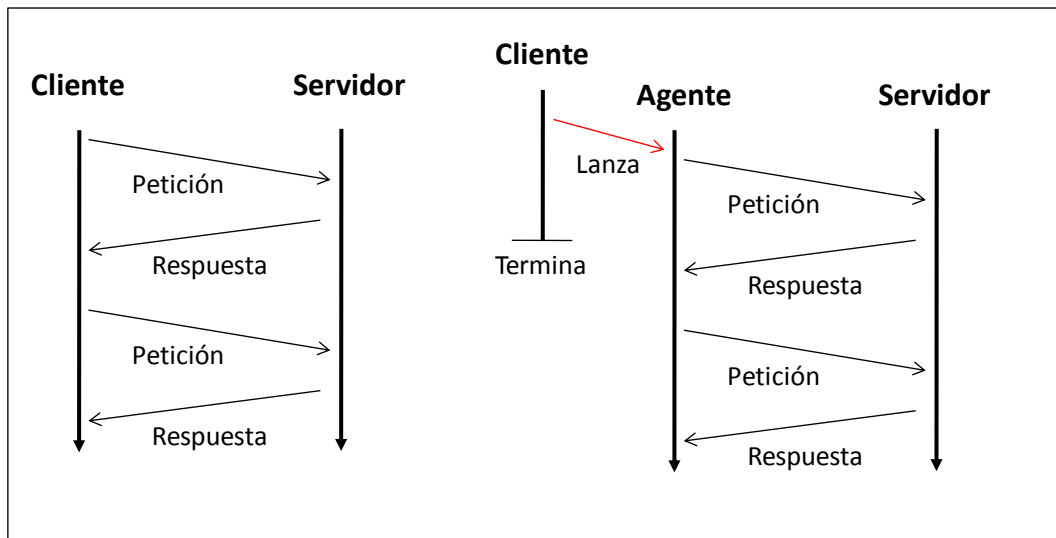


Figura 21. Ejecución asincrónica de tareas

- c) *Reducción del tráfico en la red:* Como se ha comentado anteriormente, una de las ventajas más importantes por la que esta tecnología ha tenido éxito, es debido a que es capaz de reducir una cantidad alta de tráfico en la red. Esto es debido, como se ha comentado en otros subapartados, a que la ejecución de la aplicación se desarrolla en el servidor, consiguiendo una gran eficiencia de la red, ya que se elimina el típico esquema petición/respuesta.

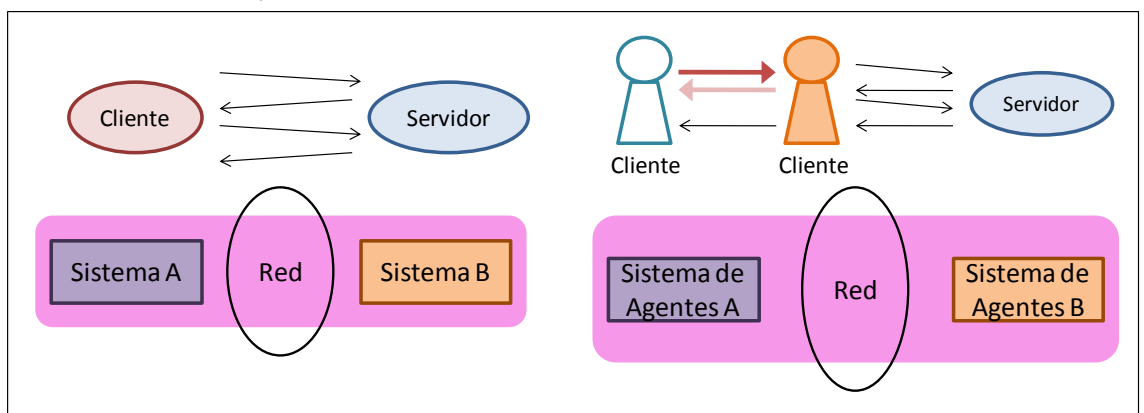


Figura 22. Reducción de tráfico en la red

Por lo tanto, las operaciones o tareas se realizan localmente en el servidor, devolviendo únicamente los resultados definitivos.

Esta labor, significa un gran filtrado de la información existente en los diferentes servidores. Es por esta razón, que en la actualidad los agentes poseen tal protagonismo, ya que la evolución que se ha producido en la red Internet ha sido tan exponencial que era necesaria una tecnología que permitiera dicho filtrado y reducción de datos en la red.

- d) *Robustos:* Debido a que el agente se va a ejecutar en el lado del servidor, consigue librarse de posibles fallos en el lado cliente y en la red. Y en el caso

de que el servidor fuese a fallar, este avisaría a todos los agentes que se están ejecutando en ese momento y, se les daría un cierto tiempo para que migraran a otro lugar.

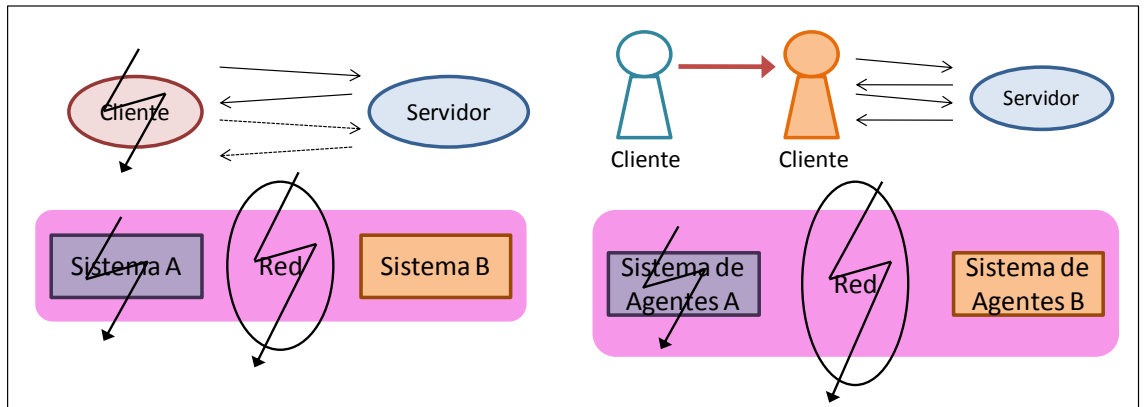


Figura 23. Robustos

- e) *Automatización del proceso de tareas distribuidas:* Debido a que estos agentes tienen la capacidad de moverse de un lugar a otro, tienen la capacidad de ejecutar una tarea en diferentes lugares para obtener el resultado deseado. Por lo tanto, es posible darles un itinerario al crearlos, para que estos viajen a los diferentes lugares donde obtendrán los datos deseados.

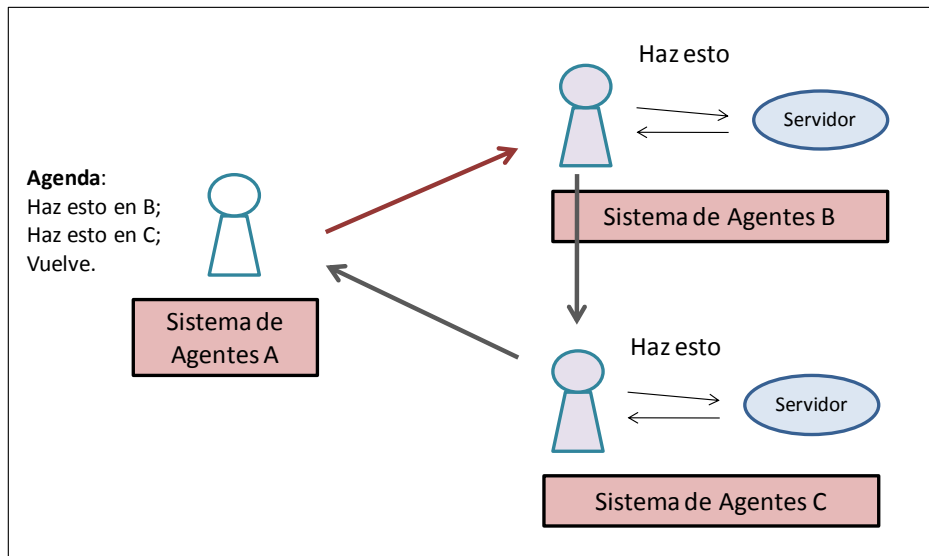


Figura 24. Automatización del proceso de tareas distribuidas

- f) *Proceso de tareas local/descentralizado:* La posibilidad de que los agentes sean móviles, como se está pudiendo apreciar, nos proporciona una infinidad de beneficios. En este caso, las tareas que un controlador tiene asignadas, las podría realizar de manera local y descentralizada, ya que con la creación de un agente móvil, este se encargaría de ejercer dichas tareas localmente en los diferentes clientes y de manera personalizada.

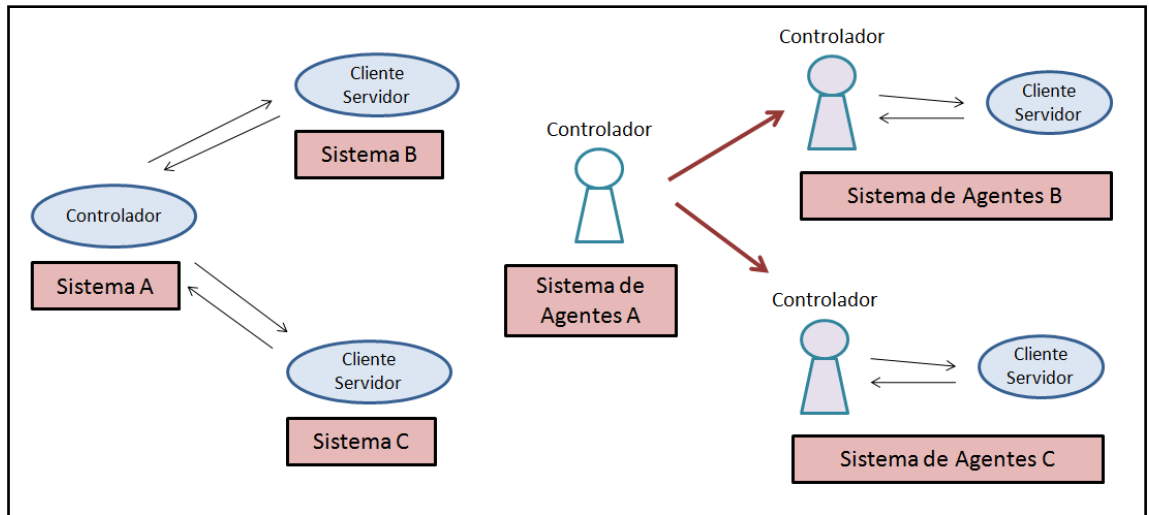


Figura 25. Proceso de tareas local/descentralizado

### 2.3.4 Aplicaciones de los agentes móviles

La tecnología de agentes móviles, gracias a las ventajas que hemos ido comentando, ha atraído el interés de muchos programadores elijan esta alternativa, dando lugar a una gama muy amplia de aplicaciones.

Algunos tipos de aplicaciones de agentes móviles son los siguientes:

- *Recolección de datos de diferentes lugares:* Los agentes móviles, como ya se ha comentado varias veces, tienen la capacidad de viajar a múltiples lugares, pudiéndoles asignar un posible itinerario. Esta capacidad, no es soportada por ninguna otra alternativa distribuida, ya que únicamente te ofrece la posibilidad de viajar a un único sitio en cada petición.
- *Monitorización:* Este tipo de aplicaciones, surge de la naturaleza asíncrona que tienen los agentes, ya que se les puede crear para realizar una determinada tarea, realizar los agentes dicha tarea sin tener la necesidad de comunicarse con el cliente y finalmente regresar cuando consiga cumplir con los objetivos. Por tanto, una aplicación típica, puede ser que un agente se le decida enviar a un servidor determinado, en el que se está publicando constantemente información reciente. De dicha información, se le restringiría a que recogiera una parte de ella muy determinada y, una vez publicada, recogerla para posteriormente mostrarla al cliente.
- *Búsqueda y filtrado:* El crecimiento de la red Internet, ha provocado que exista la necesidad de filtrar de alguna manera la información que habita en ella. Debido a esta razón y a las características de los agentes, se ha producido tal interés en esta tecnología, ya que es posible eliminar la información que no es relevante y recolectar aquella en la que un cliente está interesado.

Para finalizar, dado el grado alto de posibilidades que ofrece esta tecnología de agentes, se ha considerado en este TFM estudiar su aplicación dentro del hogar digital para que se encargue de soportar los servicios que se ofrecen.

## 2.4 Estándares de Agentes

Como sucede en otros campos y/o tecnologías de las telecomunicaciones, el número existente de aplicaciones de agentes móviles es bastante elevado. Por este motivo, y para hacer que los fabricantes y desarrolladores de plataformas de agentes no tengan problemas de interoperabilidad, varias organizaciones han llevado a cabo iniciativas con el fin de crear especificaciones de la tecnología de agentes.

En este contexto, dos de los estándares que han sido desarrollados y aceptados son MASIF y FIPA, aunque en la actualidad, el estándar con más relevancia es el de FIPA. A continuación, se describen partes de estos estándares que resultan de especial interés para el desarrollo de este TFM.

### 2.4.1 MASIF

MASIF [29], también llamado MAF (Mobile Agent Facility), es un estándar de OMG. Este estándar trata la interoperabilidad entre sistemas de agentes escritos en el mismo lenguaje, y forma parte de las utilidades definidas en CORBA. MASIF no define la estandarización de operaciones entre agentes locales, como la interpretación de agentes, serialización, ejecución, o deserialización; sino entre sistemas de agentes de diferentes tipos.

MASIF estandariza la gestión, transferencia, localización y nombrado de agentes y sistemas de agentes, y aborda la interoperabilidad en cuanto a la gestión, seguimiento y transporte de agentes; así como el nombrado de agentes y sistemas de agentes.

La gestión de agentes permite a un sistema de agentes controlar agentes de otros sistemas. MASIF aborda esta interoperabilidad mediante la definición de interfaces para llevar a cabo funciones tales como la creación, suspensión, reanudación y terminación de agentes.

El seguimiento de los agentes permite localizar agentes registrados con MAFFinders (es decir, servicios de nombres) de diferentes sistemas de agentes.

La transferencia de agentes permite que un agente de una aplicación se pueda desplazar entre sistemas de agentes de diferentes tipos (aunque con una infraestructura común). MASIF aborda esta interoperabilidad mediante la definición de métodos para la recepción de agentes y la obtención de sus clases.

La comunicación entre agentes está fuera del ámbito de MASIF, y es ampliamente dirigida por CORBA como comunicación de objetos.



#### *a) Elementos básicos de MASIF*

Un agente es un programa de ordenador que actúa de forma autónoma en nombre de una persona u organización. Cada agente tiene su propio hilo de ejecución, por lo que son capaces de realizar las tareas por iniciativa propia. Además, un agente móvil no está vinculado al sistema de agentes en el que inicia su ejecución, sino que dispone de la capacidad de transportarse a sí mismo a lo largo de la red. De este modo, cuando un agente viaja, también son transportados sus datos (atributos), su código y/o estado de ejecución.

Es necesario que cada agente tenga un nombre único, para su identificación, gestión de operaciones, y localización. Los agentes son nombrados por su autoridad, identidad, y el tipo de sistema de agentes, cuya combinación tiene un único valor.

Un sistema de agentes, identificado por su nombre y dirección, es una plataforma que puede crear, interpretar, ejecutar, transferir y terminar agentes. Al igual que un agente, un sistema de agentes está asociado a una autoridad que identifica a la persona u organización para la que actúa. En un host pueden existir uno o más sistemas de agentes.

El contexto en el que se ejecuta un agente se llama lugar. Un sistema de agentes puede contener uno o más lugares, y un lugar puede soportar uno o más agentes. Por tanto, cuando un cliente solicita la localización de un agente, éste recibe la dirección del lugar en el que se está ejecutando el agente.

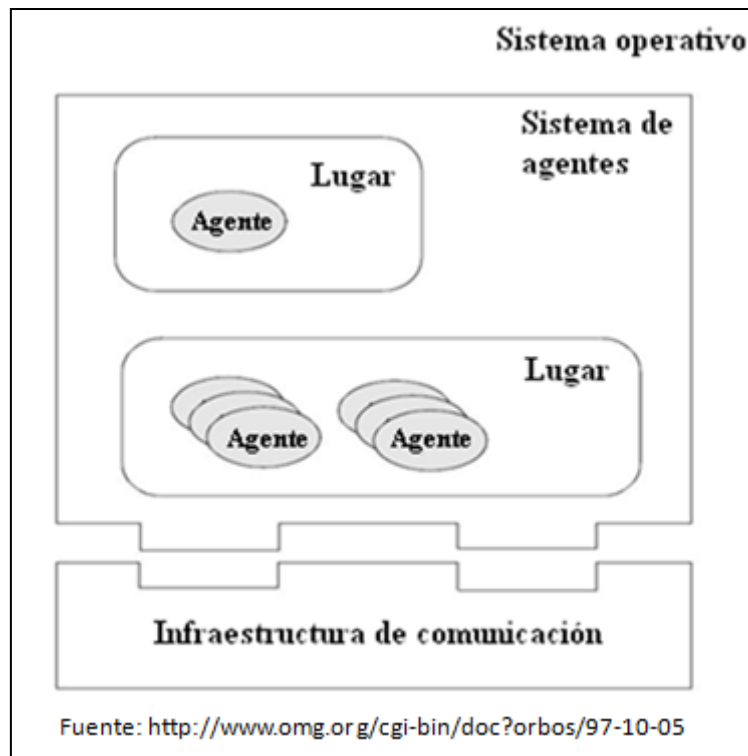


Figura 26. Sistema de agentes MASIF

Una región es un conjunto de sistemas de agentes de una misma autoridad, pero no necesariamente del mismo tipo.

#### ***b) Servicios CORBA***

Los servicios que MASIF utiliza de CORBA son: Servicio de nombres (Naming Service), servicio de ciclo de vida (Lifecycle Service), servicio de externalización (Externalization Service), servicio de seguridad (Security Service).

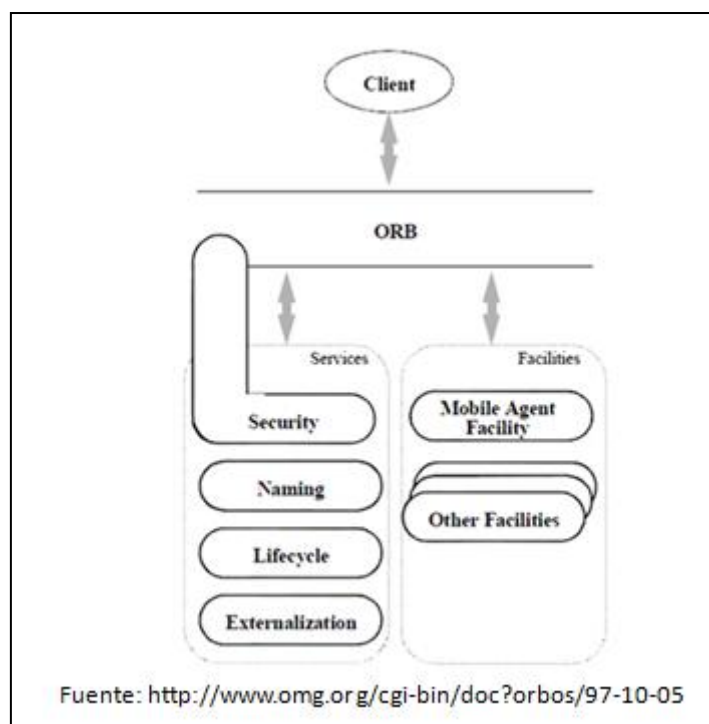


Figura 27. Servicios y utilidades CORBA

- *Servicio de nombres:* El servicio de nombres de CORBA vincula nombres a los objetos CORBA. Las aplicaciones utilizan el servicio de nombres para publicar objetos nombrados, o encontrar un objeto mediante el nombre.

MASIF define dos interfaces de objetos CORBA, *MAFAgentSystem* y *MAFFinder*, que pueden ser publicados en el servicio de nombres, si así se desea. Los agentes actúan como objetos CORBA pudiendo publicarse a sí mismos, y de este modo permitir que las aplicaciones obtengan de forma dinámica las referencias de los agentes remotos e interactuar con ellos mediante RPC CORBA. Dado que una referencia de un objeto CORBA comprende, entre otros, el nombre del host en el que reside el objeto y el número de puerto correspondiente, un agente móvil obtiene una nueva referencia después de cada migración, por lo que la referencia anterior pasa a ser inválida para el acceso de aplicaciones.

Con el fin de identificar de forma única los agentes y sistemas de agentes MASIF, se utilizan los siguientes componentes:

- *Autoridad:* Define la persona u organización a la que el agente o sistema de agentes representa.
- *Tipo de sistema de agentes:* Describe el perfil de un agente. Por ejemplo, si el tipo de un sistema de agentes es *Aglet*, el sistema de agentes está implementado por

IBM, soportando Java como lenguaje de programación de agentes y utiliza la serialización de objetos Java.

- *Identidad:* Distingue sistemas de agentes o agentes que tienen la misma autoridad y los mismos valores de tipo de sistema de agentes.

La combinación de estos tres componentes debe ser única en el contexto de MASIF.

- *Servicio de ciclo de vida:* Define las herramientas necesarias para la creación, borrado, copia y movimiento de objetos CORBA. Por tanto, los objetos CORBA definidos por MASIF, es decir, MAFAgentSystem y MAFFinder, pueden ser creados y eliminados utilizando este servicio; permitiendo también la copia o movimiento de estos objetos. Esto es útil si el host en el que se está ejecutando un sistema de agentes tiene que ser apagado o desconectado de la red.

Los agentes móviles son objetos activos con la capacidad de moverse a lo largo de la red; por tanto si un agente es representado como un objeto CORBA, es posible utilizar el servicio de ciclo de vida para su creación, borrado, copia y migración. Dado que es necesario transferir el estado del agente, el servicio de ciclo de vida debe ser combinado con el servicio de externalización de CORBA.

El servicio de ciclo de vida solo puede ser utilizado por objetos CORBA. Sin embargo, en MASIF, los agentes no tienen porqué ser objetos CORBA. Por tanto, con el fin de proporcionar una interfaz uniforme para la creación, borrado y migración de agentes basados en CORBA y no basados en CORBA, se han introducido nuevas operaciones (interfaz MAFAgentSystem).

- *Servicio de externalización:* Proporciona un mecanismo estandarizado para almacenar el estado de un objeto en un flujo de datos, y para restaurar el estado del objeto desde un flujo de datos. Un sistema de agentes utiliza este servicio cuando necesita serializar y deserializar el estado de un agente.

- *Servicio de seguridad:* Aunque la seguridad de CORBA actualmente no satisface todas las necesidades de la tecnología de agentes móviles, la implementación de MASIF debe utilizar los mecanismos de seguridad del ORB (Object Request Broker) de CORBA para satisfacer sus necesidades de seguridad. Los requisitos de seguridad para agentes y sistemas de agentes en CORBA son:

- Nombrado de agentes.

- Autenticación del cliente para la creación remota de agentes.
- Autenticación de agentes.
- Autenticación de sistemas de agentes.
- Acceso de un sistema de agentes a los resultados y credenciales de la autenticación.
- Políticas de seguridad de agentes y sistemas de agentes.
- Integridad, confidencialidad y detección de la duplicación de un agente.

Para la creación de un agente remoto, un cliente (que puede ser un agente o un programa) interactúa con el sistema de agentes destino para solicitar la creación de un agente determinado. El cliente se autentica a sí mismo en el sistema de agentes destino, estableciendo de este modo la autoridad y credenciales del nuevo agente.

Durante la transferencia de un agente, se identifica el lugar de destino; y si el sistema de agentes destino está de acuerdo, se transfiere el estado y código (si es necesario) del agente, autoridad y credenciales de seguridad.

### *c) Interfaces de MASIF*

MASIF consta de dos interfaces para lograr la interoperabilidad entre sistemas de agentes móviles. MASIF es tan simple y genérico como es posible para permitir futuros progresos en sistemas de agentes móviles.

MASIF contiene dos interfaces:

- *MAFAgentSystem*, que define las operaciones necesarias para la gestión de agentes.
- *MAFFinder*, que define las operaciones para el registro, desregistro y localización de agentes, lugares y sistemas de agentes.

- *Interfaz MAFFinder*: Se trata de un servicio de nombres que proporciona métodos para mantener nombres dinámicos y bases de datos de localización de agentes, lugares y sistemas de agentes. Puede ser compartida por varias regiones; sin embargo, por simplicidad de definición, existe un MAFFinder por cada región.

Antes de que un cliente pueda solicitar al MAFFinder que encuentre un objeto, el cliente debe obtener la referencia al MAFFinder; y para ello, el cliente puede utilizar el servicio de nombres de CORBA o el método `get_MAFFinder()` de la interfaz *MAFAgentSystem*.

- *Interfaz MAFAgentSystem*: Esta interfaz define los métodos necesarios para la gestión de agentes, tales como la creación, terminación, recepción, suspensión, restauración de agentes; así como la búsqueda del MAFFinder, agencias, listado de agentes, etc. También proporciona un conjunto básico de operaciones para la transferencia de agentes.

## 2.4.2 FIPA

### a) ¿Qué es FIPA?

FIPA (Foundation for Intelligent Physical Agents) [30] es una organización internacional sin ánimo de lucro, fundada originalmente en 1996, y registrada en Ginebra, Suiza. Se dedica a promover la industria de los agentes inteligentes mediante el desarrollo de un conjunto de estándares con la intención de promover la interoperabilidad entre agentes y sistemas de agentes heterogéneos y aplicaciones basadas en agentes.

FIPA está compuesta por compañías y universidades de telecomunicaciones e informática activas en el campo de los agentes.

Desde su fundación, FIPA ha jugado un papel crucial en el desarrollo de especificaciones de agentes y ha promovido un número de iniciativas y eventos que han contribuido al desarrollo y aceptación de la tecnología de agentes.

A lo largo de la historia de FIPA han ido surgiendo diferentes conjuntos de especificaciones: FIPA 97, FIPA 98 y FIPA 2000.

Estas especificaciones describen diferentes aspectos relativos a los agentes como: la gestión de los mismos, el lenguaje de comunicación, interacción agente-humano, movilidad, seguridad, representación de los mensajes, etc. Se pueden dividir en cinco grandes grupos:

- Aplicaciones:
  - FIPA Nomadic Application Support Specification (Especificación de soporte de aplicaciones nómadas) [FIPA00014]
  - FIPA Quality of Service Specification (Especificación de calidad de servicio) [FIPA00094]
  - FIPA Agent Software Integration Specification (Especificación de integración de software de agentes) [FIPA00079]

- FIPA Personal Travel Assistance Specification (Especificación de asistencia personal de viaje) [FIPA00080]
  - FIPA Audio-Visual Entertainment and Broadcasting Specification (Especificación de radiodifusión y entretenimiento audio-visual) [FIPA00081]
  - FIPA Network Management and Provisioning Specification (Especificación de provisionamiento y gestión de redes) [FIPA00082]
  - FIPA Personal Assistant Specification (Especificación de asistencia personal) [FIPA00083]
  - FIPA Message Buffering Service Specification (Especificación de servicio de almacenamiento de mensajes) [FIPA00092]
- Arquitectura abstracta:
    - FIPA Abstract Architecture Specification (Especificación de arquitectura abstracta) [FIPA00001]
    - FIPA Domains and Policies Specification (Especificación de políticas y dominios) [FIPA00089]
- Lenguajes de comunicación:
    - Protocolos de interacción:
      - FIPA Request Interaction Protocol Specification (Especificación del protocolo de interacción de peticiones) [FIPA00026]
      - FIPA Query Interaction Protocol Specification (Especificación del protocolo de interacción de preguntas) [FIPA00027]
      - FIPA Request When Interaction Protocol Specification (Especificación del protocolo de interacción cuando existe peticiones) [FIPA00028]
      - FIPA Contract Net Interaction Protocol Specification (Especificación del protocolo de interacción de red de contacto) [FIPA00029]
      - FIPA Iterated Contract Net Interaction Protocol Specification (Especificación del protocolo de interacción de red de contacto iterativo) [FIPA00030]
      - FIPA English Auction Interaction Protocol Specification (Especificación del protocolo de interacción de solicitud inglesa)[FIPA00031]
      - FIPA Dutch Auction Interaction Protocol Specification (Especificación del protocolo de interacción de solicitud holandesa) [FIPA00032]

- FIPA Brokering Interaction Protocol Specification (Especificación del protocolo de interacción de intermediación) [FIPA00033]
    - FIPA Recruiting Interaction Protocol Specification (Especificación del protocolo de interacción de reclutamiento) [FIPA00034]
    - FIPA Subscribe Interaction Protocol Specification (Especificación del protocolo de interacción de suscripción) [FIPA00035]
    - FIPA Propose Interaction Protocol Specification (Especificación del protocolo de interacción de propuesta) [FIPA00036]
  - Actos comunicativos:
    - FIPA Communicative Act Library Specification (Especificación de biblioteca de actuación comunicativa) [FIPA00037]
  - Lenguajes de contenido:
    - FIPA SL Content Language Specification (Especificación del lenguaje de contenido SL) [FIPA00008]
    - FIPA CCL Content Language Specification (Especificación del lenguaje de contenido CCL) [FIPA00009]
    - FIPA KIF Content Language Specification (Especificación del lenguaje de contenido KIF) [FIPA00010]
    - FIPA RDF Content Language Specification (Especificación del lenguaje de contenido RDF) [FIPA00011]
- Gestión de agentes:
  - FIPA Agent Management Specification (Especificación de gestión de agentes) [FIPA00023]
  - FIPA Agent Discovery Service Specification (Especificación del servicio de descubrimiento de agentes) [FIPA00095]
  - FIPA JXTA Discovery Middleware Specification (Especificación de descubrimiento de pasarela JXTA) [FIPA00096]
- Transporte de mensajes:
  - Representaciones de ACL:
    - FIPA ACL Message Representation in Bit-Efficient Specification (Especificación de representación en Bit-Eficiente en mensajes ACL) [FIPA00069]
    - FIPA ACL Message Representation in String Specification (Especificación de representación en cadena de caracteres de mensajes ACL) [FIPA00070]



- FIPA ACL Message Representation in XML Specification (Especificación de representación en XML de mensajes ACL) [FIPA00071]
- Representaciones de envoltorios:
  - FIPA Agent Message Transport Envelope Representation in XML Specification [FIPA00085]
  - FIPA Agent Message Transport Envelope Representation in Bit Efficient Specification [FIPA00088]
- Protocolos de transporte:
  - FIPA Agent Message Transport Protocol for IIOP Specification (Especificación del protocolo de transporte de mensajes de agentes para IIOP) [FIPA00075]
  - FIPA Agent Message Transport Protocol for HTTP Specification (Especificación del protocolo de transporte de mensajes de agentes para HTTP) [FIPA00076]
  - FIPA Agent Message Transport Protocol for WAP Specification (Especificación del protocolo de transporte de mensajes de agentes para WAP) [FIPA00084]

El conjunto de especificaciones FIPA fue oficialmente aceptado el 8 de Junio de 2005 por IEEE Computer Society, pasando de este modo a formar parte de la familia de especificaciones de normalización, con el fin de promover la tecnología de agentes y la interoperabilidad de las especificaciones con otras tecnologías (tecnologías no de agentes).

### ***b) Arquitectura Abstracta de FIPA***

En esta sección se define cómo los agentes se pueden localizar y comunicar. El propósito principal de la especificación que se ocupa de este tema es fomentar la interoperabilidad y la reutilización. Para ello, es necesario identificar elementos de la arquitectura que tienen que ser codificados, de forma que queden reflejadas las características comunes de los diferentes sistemas que utilizan distintas tecnologías para lograr algún propósito funcional.

El objetivo principal de esta Arquitectura Abstracta es conseguir el intercambio de mensajes entre agentes, que pueden utilizar diferentes mecanismos de transporte de mensajes, diferentes lenguajes de comunicación de agentes (ACL - Agent Communication Language), o diferentes lenguajes de contenido.

La Arquitectura Abstracta (Figura 28) no puede ser implementada directamente, sino que es la base para el desarrollo de especificaciones arquitecturales concretas, que describen en detalle cómo construir sistemas de agentes en términos de elementos software, tales como lenguajes de programación, interfaces de programación de aplicaciones, protocolos de red, sistemas operativos, etc.

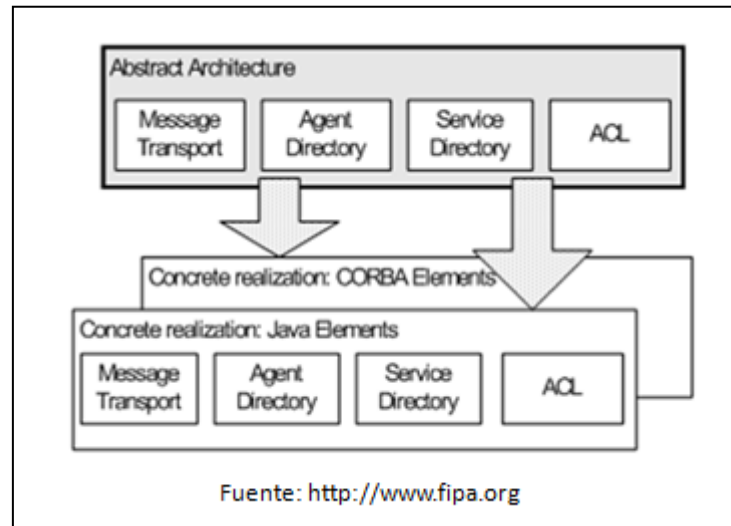


Figura 28. Implementaciones concretas de la Arquitectura Abstracta

Para que una determinada implementación de la arquitectura sea compatible con FIPA debe incluir mecanismos para el registro de agentes, localización de agentes y transferencia de mensajes entre agentes; ya que ésta define a nivel abstracto cómo dos agentes pueden localizarse y comunicarse entre ellos intercambiando mensajes.

La implementación concreta puede ser de toda la arquitectura, o sólo de un elemento. Por ejemplo (Figura 29), se podrían crear varias implementaciones concretas especificando el lenguaje de programación particular y el transporte de mensajes; y estas implementaciones podrían utilizar de forma común la implementación de un único elemento.

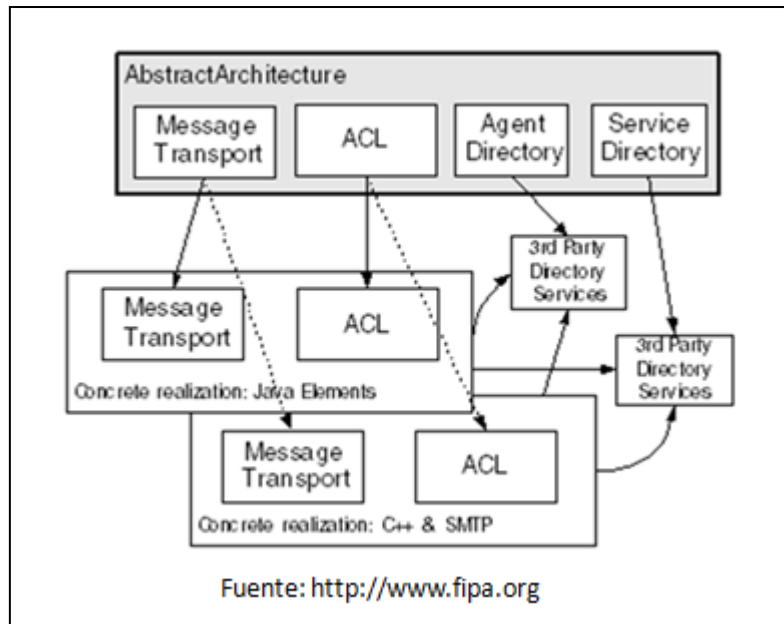


Figura 29. Implementaciones de FIPA que utilizan otras implementaciones

Cabe destacar que la Arquitectura Abstracta de FIPA define el conjunto mínimo de elementos necesarios para que una implementación concreta sea compatible con ella, permitiendo además la incorporación de nuevos elementos.

### c) Agentes y servicios

Los agentes se comunican intercambiando mensajes que representan actos de habla; estos mensajes se codifican en un lenguaje de comunicación de agentes (ACL).

La arquitectura abstracta define dos servicios proporcionados a los agentes: servicios de directorio y servicios de transporte de mensajes. Un servicio puede implementarse como agente o como software.

Los servicios de directorio proporcionan un lugar donde los agentes registran información (entradas de directorio). Esta información es utilizada por los agentes para encontrar otros con los que desean interactuar. Cada entrada de directorio se compone de:

- El nombre del agente (AID - Agent Identifier), global y único.
- Un localizador: una o más descripciones de transporte (que permiten localizar al agente), que a su vez contiene el tipo de transporte (es decir, el protocolo), la dirección de transporte para ese protocolo.

- Puede contener atributos, tales como servicios ofrecidos por el agente, coste, restricciones de uso, etc.

Cuando un agente quiere anunciarse como proveedor de un servicio, en primer lugar, se vincula a una o más direcciones de transporte, y luego se da a conocer creando una entrada de directorio y registrándola en el servicio de directorio de agentes (Figura 30).

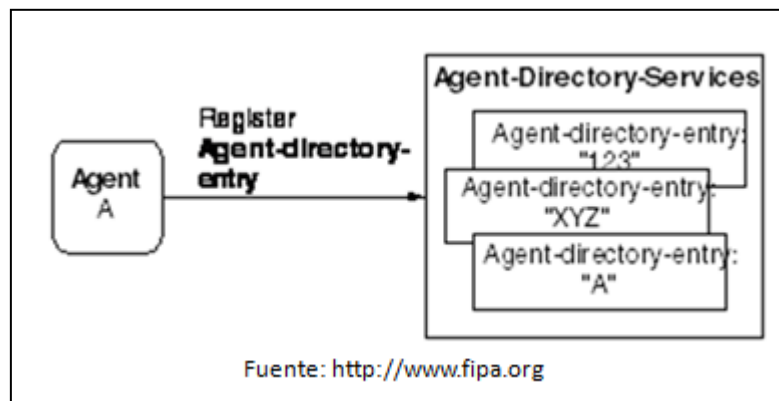


Figura 30. Un agente FIPA se registra en un servicio de directorio

#### d) Mensajes

En FIPA los agentes se comunican unos con otros mediante el envío de mensajes. Los aspectos fundamentales de la comunicación entre agentes mediante el envío de mensajes son: la estructura y el transporte del mensaje.

- *Estructura del mensaje*

Un mensaje está escrito en un lenguaje de comunicación de agentes, FIPA ACL (Figura 31). El contenido del mensaje se expresa en un lenguaje de contenido que puede hacer referencia a una ontología, la cual establece la semántica de los conceptos tratados en el contenido. Los mensajes también incluyen los nombres del emisor y el receptor, expresados como nombres de agentes, que son identificadores únicos. Cada mensaje puede tener un emisor, y ninguno (multidifusión) o más receptores. Además, un mensaje puede contener otros mensajes de forma recursiva.

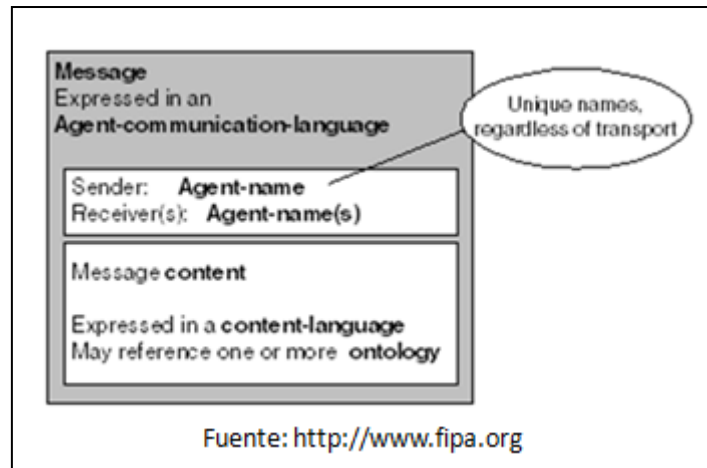


Figura 31. Estructura de un mensaje FIPA-ACL

- *Transporte del mensaje*

Al enviar un mensaje, éste se transforma utilizando una representación codificada apropiada para su transporte. El mensaje junto con las descripciones de transporte (protocolo de transporte, dirección, etc.) del emisor y receptor y quizá algunos atributos adicionales (codificación utilizada, seguridad, etc.) forman el mensaje que es enviado de un agente a otro.

Cada agente tiene una o más descripciones de transporte (relacionadas con una forma de transporte del mensaje, como IIOP o HTTP), que son utilizadas por otros agentes para enviar un mensaje. Por tanto, un agente que tenga varias descripciones de transporte, permite diferentes formas de comunicarse con otros agentes.

- *Seguridad en la transmisión de mensajes*

En la Arquitectura Abstracta de FIPA, la seguridad se proporciona mediante la validez y encriptación de los mensajes. En la validez del mensaje, los mensajes se pueden enviar de forma que se pueda identificar cualquier modificación durante su transmisión. En la encriptación del mensaje, un mensaje se envía encriptado, de forma que las entidades no autorizadas no pueden comprender el contenido del mensaje.

### ***e) Gestión de agentes FIPA***

En esta sección se especifica el modelo de referencia lógico para la gestión de agentes (creación, registro, localización, comunicación, migración y terminación de agentes).

Las entidades que forman el modelo de referencia (Figura 32) son conjuntos de capacidades lógicas (servicios) y no implica ninguna configuración física, dejando los detalles de implementación de plataformas de agentes a elección de los desarrolladores de los sistemas de agentes concretos.

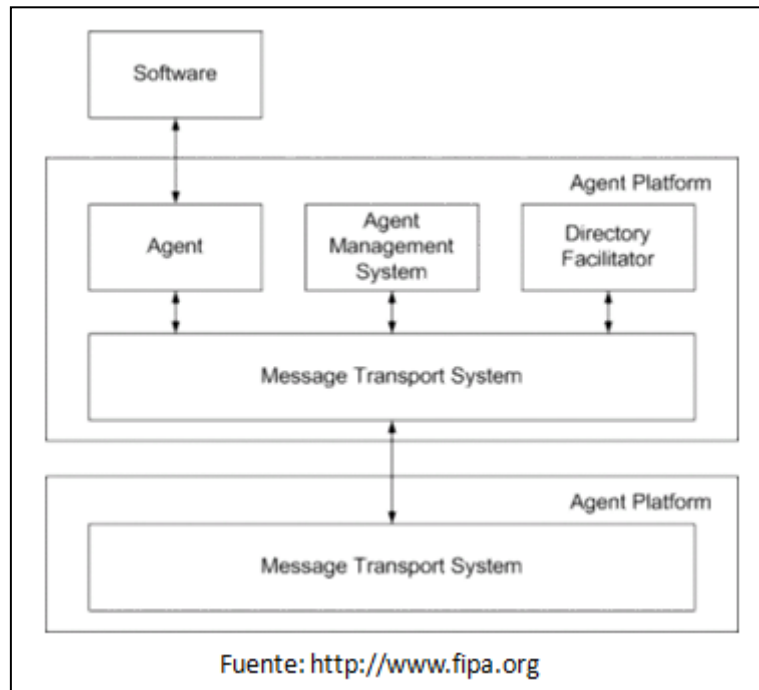


Figura 32. Modelo de referencia de gestión de agentes FIPA

- *Agente*: Es el componente básico y principal del modelo. Debe disponer de una identidad propia proporcionada por un identificador global y único (AID).
- *Facilitador de directorio (DF – Directory Facilitator)*: Se trata de un componente opcional en cualquier plataforma de agentes FIPA, pudiendo existir múltiples DFs en una plataforma. Un agente puede utilizar el DF para registrar sus servicios o para encontrar los servicios ofrecidos por otros agentes.
- *Sistema de gestión de agentes (AMS – Agent Management System)*: Se trata de un componente obligatorio en cualquier plataforma de agentes y solo puede haber uno por plataforma. Es un agente de gestión que controla el estado y el acceso a la plataforma. Mantiene un directorio de AIDs que contiene las direcciones de transporte de agentes registrados en la plataforma de agentes, permitiendo de este modo la localización de agentes a partir de sus nombres. Cada agente se debe registrar con un AMS con el fin de obtener un AID válido.

- *Servicio de transporte de mensajes (MTS – Message Transport Service)*: Es el método de comunicación por defecto entre agentes de diferentes plataformas. Todos los agentes FIPA tienen acceso al menos a un MTS.

- *Plataforma de agentes (AP – Agent Platform)*: Proporciona la infraestructura física en la que los agentes pueden ejecutarse. Una plataforma de agentes está constituida por el hardware (puede haber varios hosts), el sistema operativo, software de soporte de agentes, agentes y componentes de gestión de agentes FIPA (DF, AMS y MTS). El diseño interno de una plataforma de agentes es una cuestión de los desarrolladores del sistema de agentes y no está sujeto a estandarización en FIPA. Las plataformas de agentes y los agentes nativos de estas plataformas, ya sea porque han sido creados directamente o por un proceso de migración a la plataforma, pueden utilizar cualquier método de intercomunicación propietario; ya que FIPA solo se ocupa de cómo se lleva a cabo la comunicación entre agentes nativos de la plataforma y agentes externos.

- *Software*: Describe todo el conjunto de instrucciones ejecutables accesibles por un agente. Los agentes pueden acceder al software para añadir nuevos servicios, adquirir nuevos protocolos de comunicación, acceder a herramientas que soporten la migración, etc.

#### f) *Nombrado de agentes*

Un agente es identificado por un AID (Figura 33), que está formado por:

- El parámetro *name*, que es el identificador único que puede ser utilizado como una expresión de referencia única del agente.
- El parámetro *addresses*, que es una lista de direcciones de transporte donde se puede entregar un mensaje.
- El parámetro *resolvers*, que es una lista de direcciones con servicio de resolución de nombres.

```
(agent-identifier
 :name agent-b@bar.com
 :resolvers (sequence
 (agent-identifier
 :name ams@foo.com
 :addresses (sequence iiop://foo.com/acc))))
Fuente: http://www.fipa.org
```

Figura 33. Ejemplo de AID de un agente FIPA

La principal funcionalidad de los AID es identificar a los agentes en el envoltorio de un mensaje de transporte.

Además, añadir que la resolución de nombres es un servicio proporcionado por el AMS a través de la función search. El parámetro resolvers del AID contiene una secuencia de AID en las que el AID del agente puede ser traducido en una dirección de transporte o un conjunto de direcciones de transporte.

### g) Servicio de gestión de agentes

- DF (Directory Facilitator)

Como ya se ha mencionado anteriormente, el DF proporciona un servicio de páginas amarillas. La Figura 34 muestra dicho servicio:

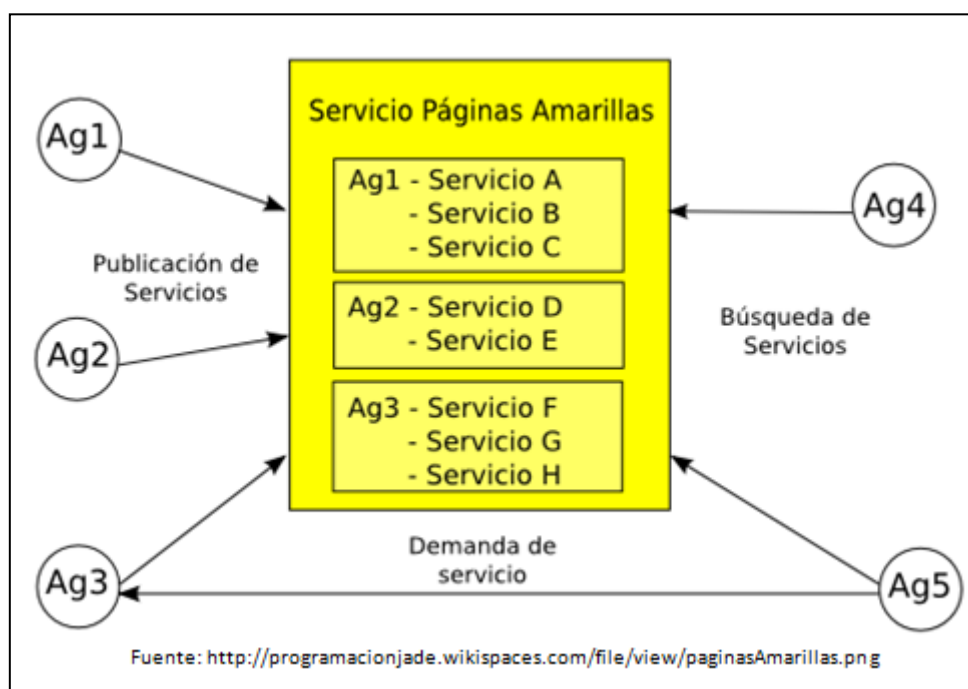


Figura 34. Estructura de un DF

En el caso de que existan DF federados, el DF constaría de un mecanismo de búsqueda que busca primero localmente y después, si está permitido, extiende la búsqueda a otros DF.



- *AMS (Agent Management System)*

El AMS proporciona un servicio de páginas blancas, siendo la máxima autoridad en toda la plataforma.

Se encarga de gestionar las operaciones que pueden llevarse a cabo en la plataforma en la que se encuentra, como la creación de agentes, borrado de agentes y supervisar la migración de los agentes a/desde la plataforma (si la plataforma de agentes soporta movilidad de agentes). Además, cada agente de la plataforma tiene asociado un ciclo de vida, que es gestionado por el AMS.

Para llevar a cabo la gestión, el AMS mantiene una lista con todos los agentes que se encuentran en la plataforma. Esta lista consta de los identificadores AID de los agentes.

#### ***h) ACC (Agent Communication Channel)***

ACC es el canal de comunicación entre agentes que proporciona el encaminamiento de mensajes entre agentes de diferentes plataformas.

El encaminamiento de mensajes entre diferentes plataformas requiere un protocolo de interoperabilidad por defecto, que es IIOP (Internet Inter-ORB Protocol). También soporta HTTP (HyperText Transfer Protocol) y WAP (Wireless Application Protocol).

#### ***i) AP (Agent Platform)***

- *Ciclo de vida de un agente en una plataforma FIPA*

Cada agente tiene un ciclo de vida que debe ser gestionado por la plataforma de agentes. El ciclo de vida de los agentes se representa mediante estados y transiciones, los cuales se muestran en la Figura 35:

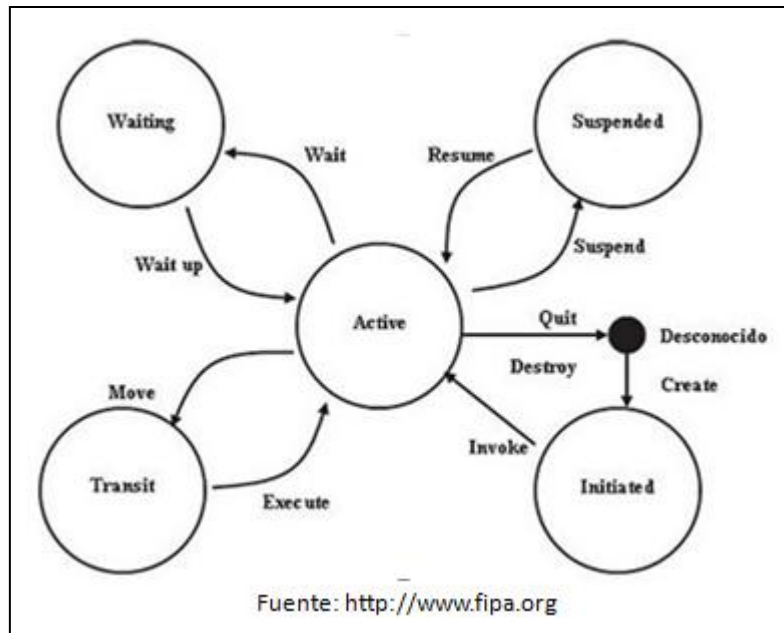


Figura 35. Ciclo de vida de un agente FIPA

A continuación se explican las responsabilidades de un AMS, en lo que respecta a la entrega de mensajes, en cada estado del ciclo de vida de un agente:

- *Active*: El MTS entrega los mensajes al agente de un modo normal.
- *Initiated/Waiting/Suspended*: El MTS almacena los mensajes en un buffer hasta que el agente vuelve al estado activo, o reenvía los mensajes a una nueva localización.
- *Transit*: El MTS almacena los mensajes en un buffer hasta que el agente vuelve a estar en el estado activo; o reenvía los mensajes a una nueva localización.
- *Unknown*: El MTS puede almacenar los mensajes o rechazarlos.

Los eventos que deben ocurrir para que un agente cambie de un estado a otro son:

- *Create*: Utilizada para la creación de un nuevo agente (el agente pasa directamente a estado iniciado).
- *Invoke*: Utilizada para la invocación o activación de un nuevo agente una vez inicializado.
- *Destroy*: Utilizada para forzar la terminación de un agente. Esta acción solo puede ser iniciada por el AMS y el agente no puede ignorarla.
- *Quit*: Utilizada para solicitar la terminación de un agente. El agente puede ignorar esta acción.

- *Suspend*: Utilizada para poner a un agente en el estado de suspendido. Esta acción puede ser iniciada por el propio agente o por el AMS.
- *Resume*: Utilizada para sacar a un agente del estado suspend. Esta acción solo puede ser iniciada por el AMS.
- *Wait*: Utilizada para poner a un agente en el estado de espera. Esta acción solo puede ser iniciada por un agente.
- *Wait up*: Utilizada para sacar a un agente del estado de espera. Esta acción solo puede ser iniciada por el AMS.
- *Move*: Utilizada para poner a un agente en el estado de transición. Esta acción solo puede ser iniciada por el propio agente.
- *Execute*: Utilizada para sacar al agente del estado de transición. Esta acción solo puede ser iniciada por el AMS.

- *Registro de los agentes*

El registro de un agente implica registrar un AID en el AMS. Cuando un agente se registra en una plataforma de agentes, el agente es registrado en el AMS, mediante la función register.

***j) Lenguaje de comunicación de agentes (FIPA ACL)***

FIPA ha desarrollado el lenguaje ACL, que se trata de un lenguaje bien definido y sin ambigüedades. La definición semántica de los mensajes ACL ha sido uno de los esfuerzos mayores dentro del estándar FIPA y otorga a este lenguaje de una gran aceptación como estándar dentro del mundo de los agentes. El elemento ACC es el que da soporte a la comunicación entre agentes empleando el lenguaje ACL.

ACL está basado en actos comunicativos que se encargan del paso de información, solicitud de la información, negociación, realización de acciones y manejo de errores.

- *Comunicación entre agentes*

El objetivo principal de los actos comunicativos entre los agentes es la solicitud de la realización de una acción a otro agente; empleándose también para solicitar operaciones a los elementos integrantes de la plataforma, como son el AMS y el DF.

Existe un conjunto amplio de actos comunicativos, cuyo detalle puede consultarse en la especificación FIPA00037 de [30].

## 2.5 Plataforma de agentes móviles para WSN

Una WSN está compuesta por nodos diminutos (muchas veces denominados motas) con sensores que interactúan mediante canales de comunicación inalámbricos. Estos nodos, cuentan con bastantes limitaciones: baja potencia, poca memoria, energía y capacidad de procesamiento limitadas, bajo ancho de banda; existiendo también la posibilidad de pérdidas de conectividad en la red.

Las WSN están teniendo cada vez mayor importancia debido al gran número de posibles aplicaciones que ofrecen, como el entorno militar, el hogar digital, medioambiente, biomedicina, etc. Una WSN puede ser desplegada en entornos de difícil acceso, de modo que la recarga o sustitución de los sensores sería muy difícil o incluso imposible. Por tanto, los sensores deben ser robustos y autónomos, reduciendo de este modo la dependencia con cualquier otro sensor, y además, las aplicaciones que se ejecuten en éstos deben consumir la mínima cantidad de recursos, prolongando así su tiempo de vida lo máximo posible.

Desde este punto de vista, parece obvio que la tecnología de agentes móviles también se haya llegado a aplicar en las WSN, ya que como se ha mencionado en anteriores apartados, su utilización supone grandes ventajas, como la reprogramación de los nodos desplegados o la reducción del tráfico en la red, entre otras. Sin embargo, para que una plataforma de agentes se pueda aplicar en una WSN, dicha plataforma debe ser capaz de cumplir los requisitos mencionados anteriormente.

Hasta la fecha, el uso de agentes sobre WSN se ha llevado a cabo en diversos campos, entre los que se pueden destacar algunos proyectos, como el que llevó a cabo DARPA en el entorno militar, así como el que llevaron a cabo los investigadores de la UCLA [31] en el entorno forestal.

Cabe destacar, que los agentes móviles en las WSN es un tema con una gran proyección científica, existiendo un número considerable de plataformas, tales como Agilla [32] [33] [34], In-Motes [35], etc. Sin embargo, estas plataformas carecen de compatibilidad con los estándares de agentes existentes.

En este TFM se estudiarán las plataformas Agilla, que como se verá a continuación, cuenta con varios inconvenientes, tales como que el mecanismo de reacciones sólo funciona en el espacio de tuplas local y sólo permite la comunicación con vecinos a un solo salto, e In-Motes. Por último, se expone una arquitectura middleware compatible con el estándar FIPA, que soluciona estos problemas.

### 2.5.1 Aguilla

Agilla es el primer middleware de agentes móviles para WSN. Proporciona un estilo de programación de agentes móviles, que permite al agente migrar proactivamente su código a través de la red. Dentro de una red, los agentes móviles pueden compartir los recursos de un único nodo, es decir, varios agentes pueden coexistir en un mismo nodo, maximizando, de este modo, la utilidad de la WSN. El número de agentes que puede ejecutar sus aplicaciones en cada nodo es variable y viene determinado por las limitaciones de memoria.

Debido a que el middleware Agilla no cumple con las especificaciones del estándar FIPA, él mismo debe elaborar sus propios mecanismos para soportar comunicaciones fiables entre los agentes. Por este motivo, en puntos sucesivos en este TFM, se incluye una propuesta de la Universidad de Darmstadt (Alemania) para mejorar este middleware y hacerlo compatible con plataformas que cumplan con dicho estándar.

Este middleware ha sido probado sobre motas Mica2 [36], MicaZ [37] y Tmote Sky [38].

#### *a) Modelo Agilla*

El middleware de Agilla se ha implementado en su totalidad sobre TinyOS [39] y proporciona dos recursos fundamentales en cada nodo: una lista con todos los vecinos del nodo y un espacio de tuplas.

También es importante dentro de este modelo, el ciclo de vida de los agentes y el direccionamiento de los nodos.

- *Espacio de tuplas*

Dado que una aplicación de Agilla contiene numerosos agentes, y de diferentes tipos, dispersos a lo largo de la red, deben existir mecanismos de coordinación que permitan a estos agentes comunicarse. Para ello, los diseñadores de Agilla han desarrollado un mecanismo de memoria compartida, conocido con el nombre de espacio de tuplas. En este espacio de tuplas, los agentes pueden insertar tuplas y, del mismo modo, acceder a ellas. Cada tupla dentro de cada nodo contiene información relevante para la comunicación. Si un agente que creó una tupla muere, dicha tupla permanece en el nodo en el espacio de tuplas y podrá ser accedida por el resto de agentes que se encuentren en el mismo nodo y, en caso de agentes de nodos diferentes, de forma remota.

Una tupla es un conjunto ordenado de campos donde cada campo tiene un tipo y un valor. Los diferentes tipos aceptados en el espacio de tuplas son enteros, localizaciones y lecturas de los sensores, entre otros.

Para acceder a estas tuplas se requiere el uso de patrones o plantillas. Estas plantillas son un conjunto ordenado de campos de forma similar a una tupla, pero con la salvedad de que en sus campos podrían contener comodines que coinciden con el tipo. Para que un agente pueda extraer una tupla del espacio de tuplas, deberá insertar una plantilla que coincida con la tupla. Para que ambas coincidan tienen que tener el mismo número de campos, y cada campo de la tupla debe coincidir con un campo de la plantilla.

El espacio de tuplas proporciona a los agentes una forma conveniente de descubrir su contexto. Por ejemplo, dado un nodo determinado, con sensores de diferentes tipos, un agente podrá conocer los tipos de sensores disponibles en él, gracias a la inserción de tuplas especiales dentro de dicho espacio de tuplas. En el caso que un nodo tenga un termómetro, se insertará una “tupla-temperatura”. Otros datos de contexto almacenados en el espacio de tuplas son la localización y el número de agentes localizados.

Las operaciones básicas sobre el espacio de tuplas son de dos tipos:

- *Bloqueantes*: la ejecución se bloquea hasta que exista coincidencia con alguna tupla.
- *No bloqueantes*

Para que un agente pueda indicar a un nodo cuales son las tuplas en las que está interesado, Agilla incluye el mecanismo de las reacciones sobre el espacio de tuplas. Mediante este mecanismo, los agentes insertan una reacción (una tupla especial), y cuando una tupla coincidente sea insertada en el espacio de tuplas, le será notificado al agente permitiéndole responder. Sin este mecanismo, a los agentes no se le notificarían las tuplas insertadas, pudiéndose quedar bloqueados esperando a que apareciera. Este mecanismo de reacciones solo está disponible sobre un espacio de tuplas local, es decir, un agente solo podrá reaccionar a tuplas insertadas en su espacio de tuplas local y no sobre los espacio de tuplas de nodos remotos.

- *Lista de vecinos*

Contiene las direcciones de todos los nodos vecinos a un solo salto. Esto es necesario, ya que un agente cuando quiere moverse o clonarse necesita la dirección destino correspondiente a la migración. Agilla permite el descubrimiento de los nodos

vecinos mediante el uso de balizas. Esta lista de nodos vecinos se actualiza de forma continuada para solventar los posibles cambios en la infraestructura de la red.

La siguiente figura es un esquema del modelo descrito anteriormente:

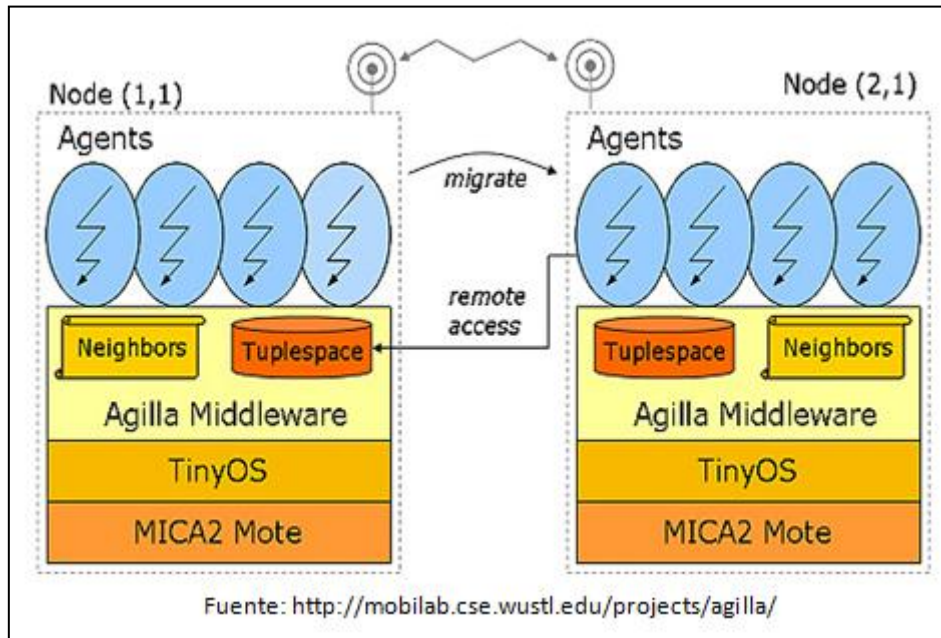


Figura 36. Modelo de agentes de Agilla

- *Ciclo de vida*

El ciclo de vida de un agente comienza cuando el agente es introducido en la red, ya sea a través del usuario o clonado por otro agente que ya se encuentra dentro de dicha red.

Cada agente contiene su propia arquitectura de pila (memoria de instrucciones y de datos, contador de programa, etc.), que será explicada en la siguiente sección de este TFM. Cuando un agente migra, se lleva consigo su código y su estado, pero no su espacio de tuplas. Cuando un agente se mueve, se lleva su estado y código propios y reanuda la ejecución en el nodo destino, sin embargo, cuando el agente se clona, se copia su estado y su código al nuevo nodo y se reanuda su ejecución en ambos. La primera letra de la instrucción de migración indica si ésta es débil o fuerte (w=débil, s=fuerte). En una operación débil solo se transfiere el código, la pila del programa es reseteada y el agente reanuda su ejecución desde el principio. En una operación de migración fuerte, se transfiere tanto el código como la pila del programa y el agente reanuda su ejecución en el punto donde lo dejó.

Un agente puede moverse y clonarse él mismo a cualquier nodo, a pesar del número de saltos. Cuando un agente completa su tarea muere, liberando así sus recursos para que puedan usarlos otros agentes. Un agente muere a través de la instrucción halt.

- *Direccionamiento*

Dentro de la red, los nodos se encuentran localizados mediante coordenadas de tipo (x, y). Es decir, aunque los agentes tengan un identificador que los identifique, a la hora de clonación, no se clonaría el agente con identificador = X, sino que el agente se clonaría a un nodo con localización, por ejemplo, (3, 3), tal y como se muestra en la siguiente figura:

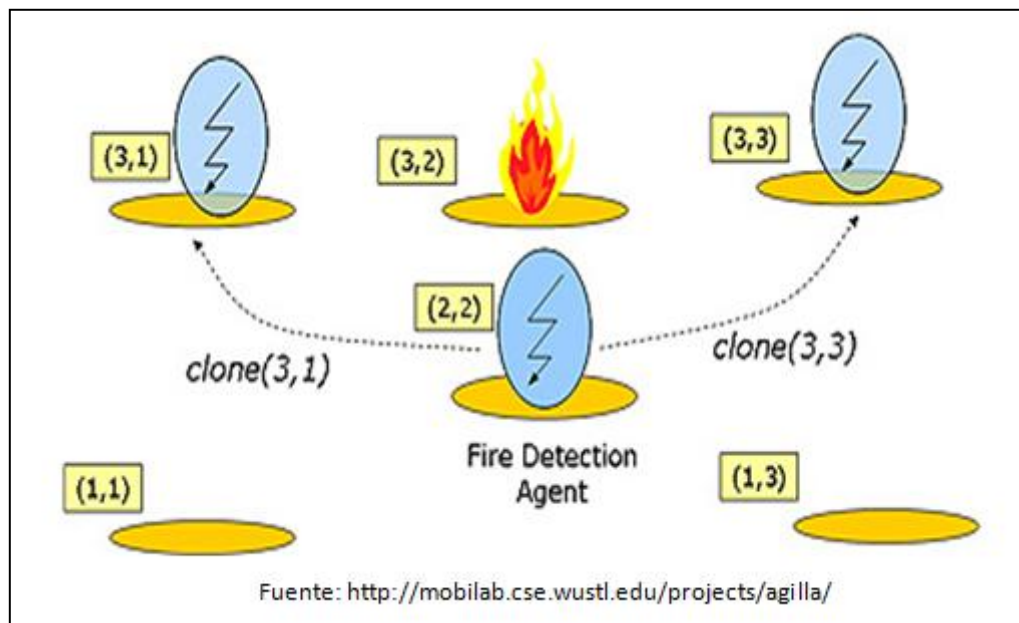


Figura 37. Direccionamiento en Agilla

**b) Arquitectura de Agilla**

La arquitectura de Agilla es la que se muestra a continuación:



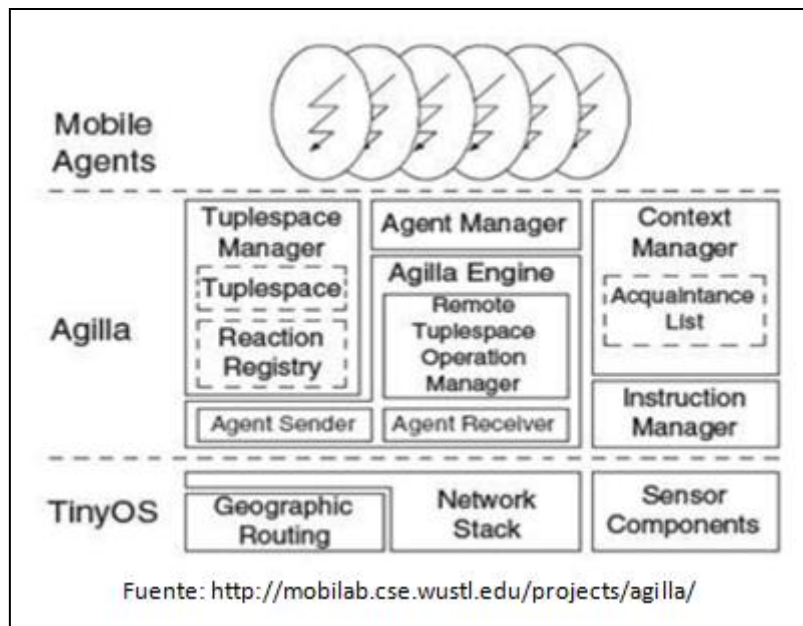


Figura 38. Arquitectura de Agilla

Tal y como se puede apreciar en la imagen existen en tres niveles:

- El nivel más alto correspondiente a los agentes.
- El nivel intermedio que contiene los componentes del middleware del corazón de Agilla.
- El nivel inferior que se corresponde con el sistema operativo TinyOS. De este nivel no se especifica estructura porque no es específico de Agilla.

- *Nivel intermedio: middleware Agilla*

Consta de un gestor de agentes, un gestor de contexto, un gestor de instrucciones, un gestor del espacio de tuplas y una máquina que los dirige (máquina Agilla).

- *Gestor de agentes:* El gestor de agentes es el encargado de los posibles contextos de un agente. Es el responsable de la asignación de memoria a un agente cuando llega y de la des-asignación cuando deja el nodo o muere. También es el que decide cuando se ejecuta un agente, y se lo notifica a la máquina Agilla cuando sucede. Por defecto, el gestor de agentes puede manejar hasta 4 agentes. Esto es fácilmente configurable y principalmente está limitado por la velocidad del procesador y la cantidad de memoria disponible.
- *Gestor de contexto:* El gestor de contexto determina la localización de un agente así como la de sus vecinos. Utiliza balizas para descubrir vecinos

y almacena las localizaciones de sus vecinos en una lista conocida y accesible para los agentes. Para un agente conocer su localización y la de sus vecinos es algo vital.

- *Gestor de instrucciones:* La asignación fija de memoria en el uso de los agentes es poco eficaz, ya que, los agentes no usan toda la memoria asignada y además limita el tamaño máximo del programa de un agente. Por este motivo, Agilla incluye un mecanismo de asignación dinámica de memoria a los agentes que es implementado por el gestor de instrucciones. Cuando un agente llega a un determinado nodo requiere una cantidad de memoria para ejecutar sus instrucciones. Por defecto, este gestor asigna a cada agente un determinado número de bloques de 22 bytes necesarios para almacenar su código. Otra función de este gestor es la de obtener la siguiente instrucción a ejecutar por un agente.
- *Gestor del espacio de tuplas:* El gestor del espacio de tuplas usa todas las operaciones del espacio de tuplas no bloqueantes y se encarga de gestionar los contenidos del espacio local de tuplas y los registros de reacciones. La asignación de memoria para las tuplas también se realiza de forma dinámica. Por defecto, son asignados 600 bytes y una tupla podría contener hasta 25 bytes en el valor de los campos. Cuando se elimina una tupla, las siguientes se desplazan hacia adelante en las posiciones de memoria. Este gestor recuerda las reacciones registradas por cada agente mediante el almacenamiento de estas dentro del registro de reacciones. Cuando una tupla es registrada, se comprueba el registro por si existe alguna coincidencia. Si se da este caso, el gestor del espacio de tuplas se lo notificará al gestor de agentes, que actualizará el contador del programa del agente determinado para que ejecute el código correspondiente. Durante la migración, el gestor del espacio de tuplas empaqueta todas las reacciones de un agente, por eso pueden ser transferidas con el agente. Cuando un agente llega a un nodo nuevo, automáticamente se le restauran todas sus reacciones.
- *Máquina de Agilla:* La máquina de Agilla es el núcleo de la máquina virtual que controla la ejecución concurrente de todos los agentes en un mismo nodo. Esta máquina implementa una política de programación round-robin, donde cada agente puede ejecutar un número fijo de instrucciones antes de interrumpir su contexto. El número por defecto son 4 instrucciones. Además maneja las entradas y salidas de los agentes. Cuando un agente migra, no puede ser enviado en un único mensaje.

- *Nivel superior: arquitectura de los agentes*

Cada agente mantiene una arquitectura de pila de forma independiente al resto de los agentes. Esta arquitectura se muestra en la siguiente figura:

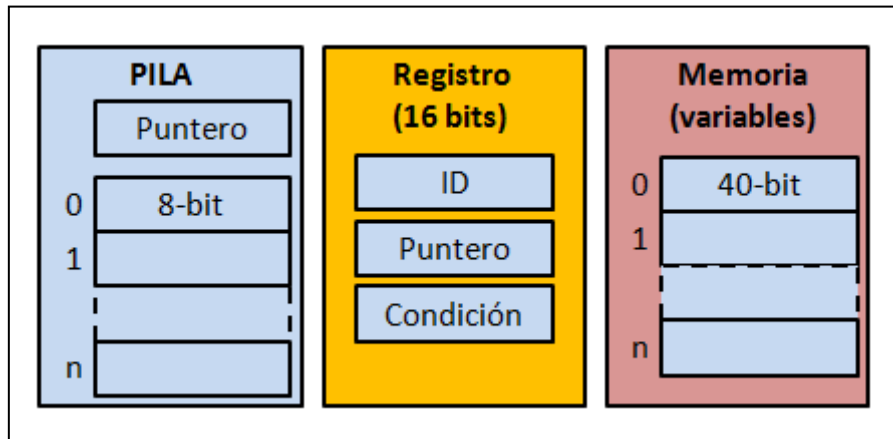


Figura 39. Arquitectura de pila de un agente en Agilla

Consta de una pila, varios registros y una zona de memoria. El motivo de usar una arquitectura de pila es que la mayoría de las instrucciones pueden ser de un único byte.

La pila es lo que conocemos como pila de ejecución del programa, es decir, contiene todas las instrucciones en orden de la aplicación que el agente debe ejecutar.

Los tres registros existentes son de 16 bytes. El primero de ellos contiene el ID del agente. El ID es único para cada agente y nunca cambia a pesar de las operaciones de migración. En el caso de clonación, al nuevo agente se le asignará un ID diferente. El registro PC es el contador del programa, es decir, contiene la dirección de la siguiente instrucción a ejecutar. Es modificado a través del salto de instrucciones y es usado por el gestor de instrucciones cuando va a buscar la siguiente instrucción a ejecutar. Cuando se da una reacción, este registro apuntará a la primera instrucción del código de la reacción. Además, el valor del registro antes de la reacción se almacena en la pila, permitiendo de este modo, reanudar la ejecución del programa en el punto donde se disparó la reacción. Por último, el registro Condición almacena el estado de la ejecución.

La memoria es un área de almacenamiento que permite a un agente almacenar hasta 12 variables. Para acceder a esta área se utilizan las operaciones setvar y getvar.

### 2.5.2 In-Motes

Ya que los agentes móviles pueden moverse inteligentemente o clonarse ellos mismo a localizaciones deseadas en respuesta a cambios en el entorno, se pensó en In-Motes como un middleware de agentes móviles inteligentes para WSN, que permite a dichos agentes ejecutarse en paralelo, de forma independiente. Al igual que Agilla fue implementado sobre el sistema operativo TinyOS y sobre motas MICA2.

Este middleware, basado en el modelo de agentes de Agilla, permite a los usuarios inyectar agentes dentro de una red, y del mismo modo, hereda de dicho modelo el espacio de tuplas (sistema federado) a través del cual se comunican los agentes, captando así los cambios en el entorno; y la gestión dinámica de memoria para los agentes, instrucciones y espacios de tuplas, debido a que el sistema TinyOS únicamente proporciona asignación estática de memoria.

Además, para minimizar el impacto de la pérdida de mensajes, muy común en agentes móviles, In-Motes a la hora de migrar agentes dentro de la red, los divide en paquetes diminutos que son retransmitidos a los nodos destino.

En cuanto al direccionamiento, se realiza a través de un identificador asignado a los nodos. Este sistema es más razonable que usar direcciones geográficas, ya que, en las aplicaciones dinámicas de WSN, al llevarse a cabo en entornos reales, las localizaciones de los nodos pueden cambiar a lo largo de su ciclo de vida o activación de la red.

#### *a) Arquitectura In-Motes*

Como se comentó anteriormente, los agentes son capaces de coordinar sus acciones a través de un espacio de tuplas, pero además, In-Motes añade un conjunto de reglas de comportamiento y comunicación de agentes basados es el siguiente modelo facilitador:

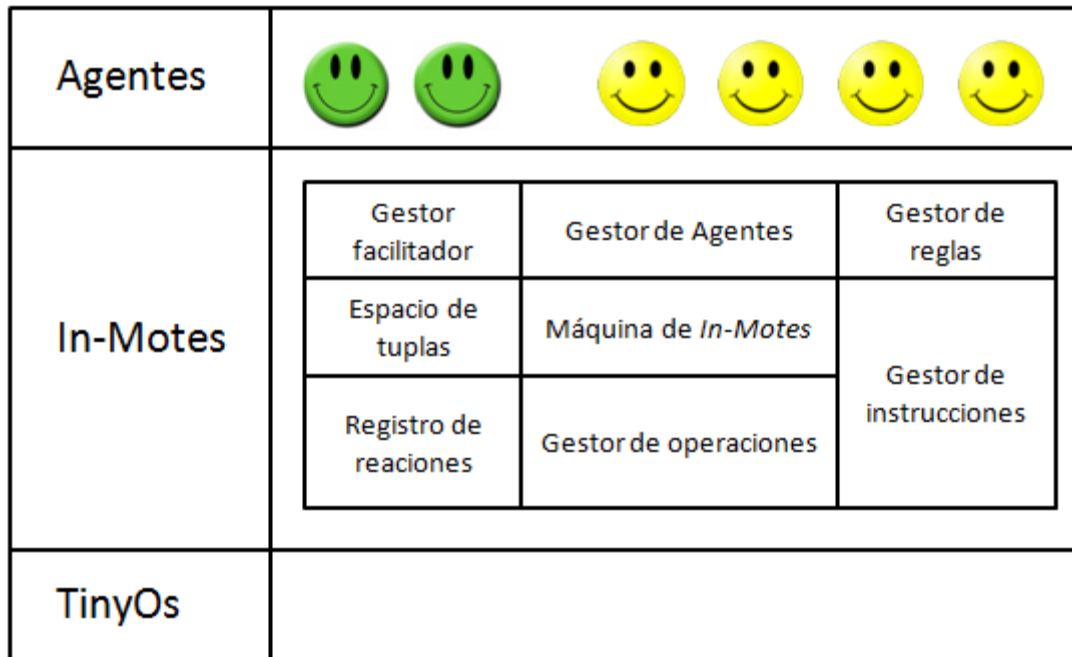


Figura 40. Arquitectura de In-Motes

Tal y como se ve en la imagen anterior, la arquitectura de In-Motes se divide en tres niveles: el sistema operativo TinyOS, el propio nivel In-Motes y un nivel facilitador de agentes.

El nivel facilitador de agentes proporciona dos tipos de agentes: agentes maestros y agentes esclavos.

El gestor de agentes es el responsable de mantener el contexto de los agentes, tanto maestros como esclavos. Cuando un agente, sea del tipo de que sea, llega a una mota, este gestor le asigna una cantidad de memoria específica para poder ejecutarse, y le comunica este hecho a la máquina In-Motes. Debido a limitaciones del hardware, el gestor de agentes solo es capaz de manejar dos agentes maestros.

El gestor de reglas se encarga de mantener las reglas de comportamiento de cada agente maestro. Cada vez que un agente de este tipo está preparado para su ejecución, este gestor informa al gestor de instrucciones y crea una lista de reglas de comportamiento para dicho agente, que son mantenidas hasta que termina su ejecución.

El componente gestor de instrucciones comprueba de forma constante la llegada de agentes a la red para asignarles recursos para ejecutar instrucciones específicas. Además, recupera la siguiente instrucción a ejecutar del código del agente.

El gestor facilitador tiene el objetivo de coordinar todos los espacios de tuplas y operaciones de reacción. En cuanto a los espacios de tuplas, asigna memoria de forma dinámica para todas las tuplas insertadas. Cada tupla pueden contener hasta cuatro campos, con el fin de ajustarse a los 27 bytes de carga útil de un mensaje del nivel TinyOS. En cuanto a las operaciones de reacción, almacena todas las reacciones registradas a través de un agente esclavo. Durante la inserción de una tupla, este gestor comprueba los registros de reacciones en busca de coincidencias. Si se obtuviera alguna coincidencia con la tupla insertada, les notificaría este hecho al gestor de instrucciones y al gestor de agentes, para que el propio agente reciba dicho aviso.

Por último, la máquina In-Motes, que actúa como una máquina virtual sobre el sistema operativo TinyOS, controlando tanto la ejecución, como la entrada y salida de los agentes. Cuando un agente entra en un estado de suspensión (wait o sleep), la máquina In-Motes apaga su contexto de ejecución. En el caso de pérdida de algún paquete, la máquina lo retransmitirá hasta un máximo de seis intentos, sino se consiguiera éxito, la operación se pararía. Actualmente, el tiempo de retransmisión de un paquete, está sujeto a investigaciones.

Debido a las limitaciones de energía y ancho de banda de los nodos de la red, no existe un espacio de tuplas global para todos los nodos, sino que cada nodo posee un número de espacios de tuplas locales únicos. Un espacio de tuplas de un nodo maestro es compartido por todos sus nodos locales esclavos y es accesible de forma remota gracias a instrucciones especiales. Para acceder a un espacio de tuplas remoto, solo se requiere el envío de dos mensajes, petición y respuesta, mientras que si un nodo maestro quiere consultar los espacios de tuplas de sus nodos esclavos utilizará un mensaje multicast.

#### ***b) La aplicación In-Motes Bins***

Para mostrar la funcionalidad de In-Motes se va a describir una de sus aplicaciones [35]. Se desplegaron 10 motas MICA2 localizadas en varios sitios de un almacén de vinos; estas motas son capaces de detectar la luz y la temperatura a través del uso de sensores MTS310CA; una estación base (un portátil en este caso) conectado a una interfaz de unión con la motas situado en una zona determinada del almacén sirve como punto de agregación de agentes. La siguiente figura muestra una descripción de esta infraestructura desplegada:

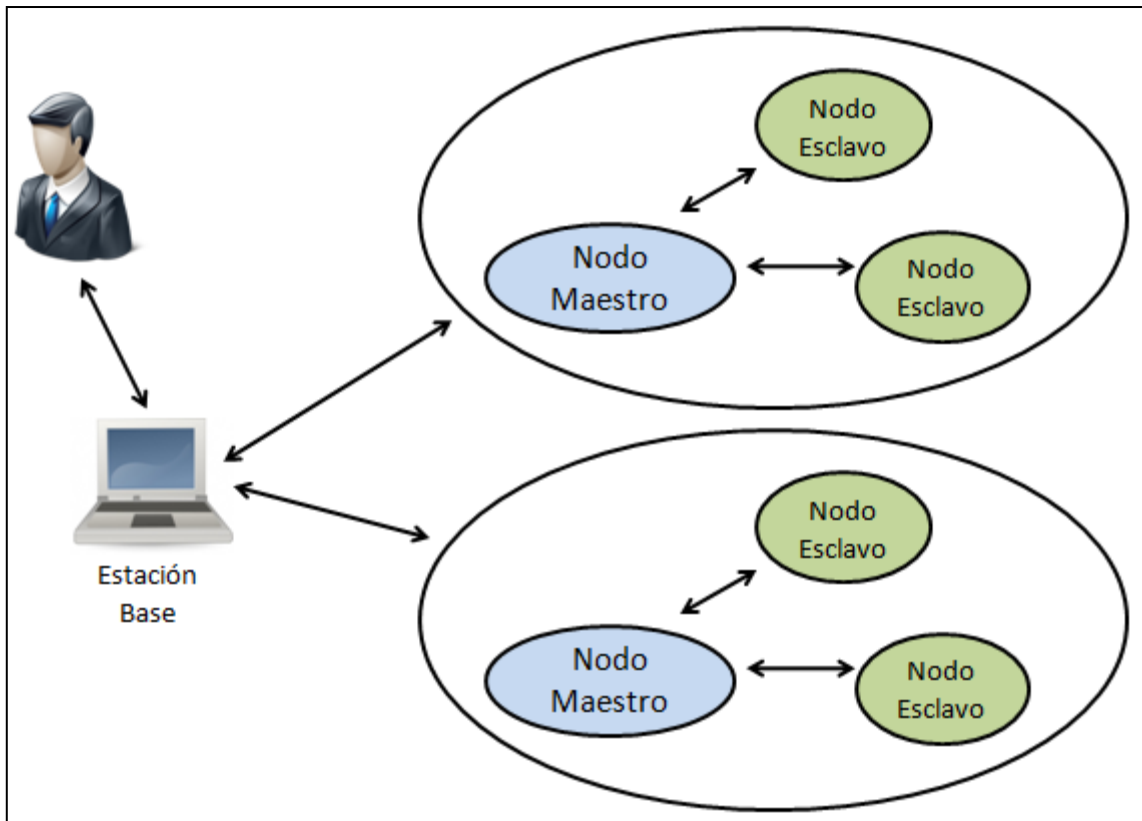


Figura 41. In-Motes Bins

Con el fin de comprender de una forma más completa el funcionamiento de esta arquitectura de agentes, a continuación se describe el proceso que se lleva a cabo en el almacén de vinos anteriormente mencionado, para mantener los productos en buen estado.

Antes de comenzar con la explicación, comentar que cuando se habla de nodos maestro o esclavos, se refiere a nodos en los cuales residen agentes maestros o esclavos respectivamente.

En la primera etapa de la aplicación se lleva a cabo el despliegue del nodo maestro, que es un nodo facilitador de nodos/agentes esclavos. Éste está continuamente ejecutando un proceso de captura, mediante el cual registra las reacciones correspondientes a sus agentes esclavos. Además, mantiene un contador que se incrementa cada vez que se registra una reacción a una captura en el nodo. Cuando este contador alcanza su máximo valor, significa que el nodo maestro ha capturado el número máximo de agentes esclavos que es capaz de manejar, por lo que se clonará y migrará al siguiente nodo disponible, asignando un nuevo agente maestro a los espacios de tuplas y reacciones y comenzando de nuevo el proceso de captura.

Cada vez que un agente esclavo es capturado, el agente maestro inserta una tupla en blanco en el espacio de tuplas local, activando así tres leds en un dispositivo flash durante tres segundos, con el fin de proporcionar información visual del estado de la red.

Durante la segunda etapa, la aplicación trata las peticiones del usuario. A los agentes esclavos se les puede asignar dos tipos diferentes de trabajos mediante la inserción de dos tipos de tuplas distintas: aquellas que contengan la cadena "pho" indican que el trabajo del agente está destinado a la medida de luz, mientras que, aquellas que contengan la cadena "temp" a la medida de la temperatura. Si además, las tuplas contienen la cadena "static", el agente ejecutará una única petición de medida y después morirá (agente estático), mientras que si esta cadena no es insertada en la tupla, el agente será dinámico y podrá realizar múltiples medidas, respondiendo a los cambios en los parámetros definidos por el usuario.

Una vez definido el trabajo del agente y en el caso de que sea un agente estático, éste insertará una única tupla de trabajo dentro del espacio de tuplas del nodo maestro. Cada nodo esclavo, transmitirá sus lecturas de temperatura o luz a su nodo maestro y este reportará dichas lecturas al usuario a través de la interfaz de unión con la estación base. Estos agentes se redistribuyen de forma automática cada 30 minutos.

Los agentes estáticos no proporcionan toma de decisión local, mientras que los agentes dinámicos pueden decidir en base a unos parámetros críticos específicos si una medida debe ser reportada al usuario o no. Estos agentes a su llegada al nodo insertan en el espacio de tuplas una tupla que contienen una cadena que se corresponden con el parámetro umbral específico del trabajo. Del mismo modo que ocurre con los agentes estáticos, se toman medidas cada treinta minutos, y en base al valor del parámetro umbral y medido, el nodo maestro reportará o no las lecturas al usuario. Estos agentes dinámicos fueron creados con el fin de proporcionar una mayor energía a las motas, transmitiendo sus medidas a través de la eliminación de los agentes redistribuidos.

### 2.5.3 Arquitectura para WSN basada en FIPA

En esta propuesta de la Universidad de Darmstadt [40], una WSN está dividida en clusters, donde cada uno de ellos está gestionado por un cabeza de cluster.

Esta propuesta consta de:

- *Agentes de gestión*: Son aquellos agentes que forman parte del middleware y realizan tareas de gestión: CMA (Cluster Management Agent) y CDA (Cluster Directory Agent).



- *Agentes de aplicación:* Son aquellos agentes que forman parte de la capa de aplicación y realizan tareas asignadas por el usuario.
- *Gestor de la movilidad del agente.*
- *Subsistema de comunicación, proporcionando servicios de comunicación de mensajes entre los agentes:* NLM (Neighbor List Manager) y LSDAC (Local Sensor Data Acquisition Component).

La arquitectura de este modelo es la siguiente:

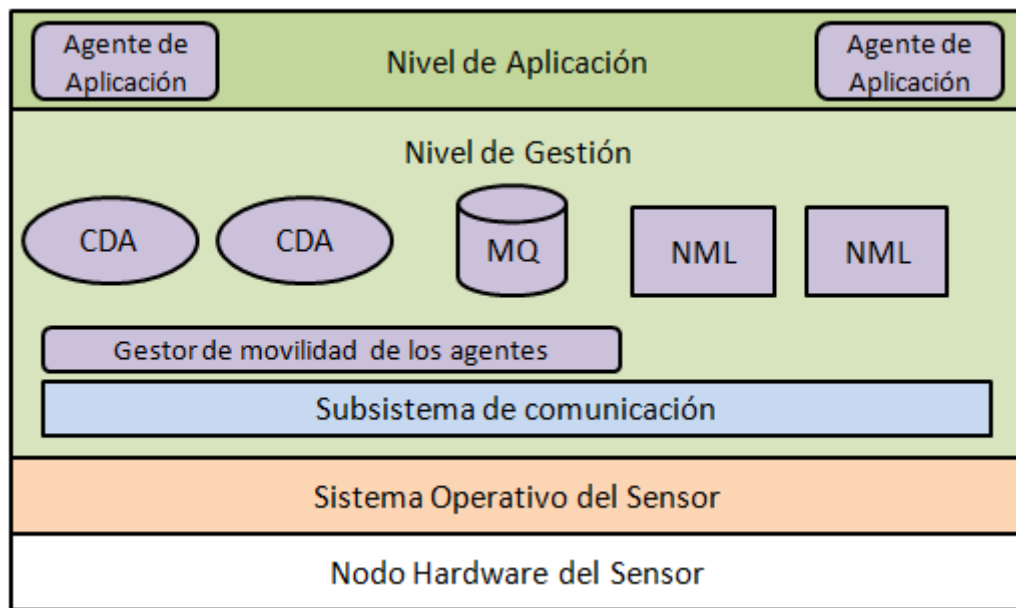


Figura 42. Arquitectura de agentes móviles para WSN basada en FIPA

Los elementos que forman parte de esta arquitectura se describen en los apartados siguientes.

**a) Componentes middleware en los nodos cabeza de cluster**

- *CMA:* Es el agente gestor del cluster, que se encarga de controlarlo. Todos los agentes de aplicación que se encuentran presentes en su cluster son registrados aquí. Además, controla su ciclo de vida y puede crear y activar o terminar cualquiera de ellos; encargándose también de asegurar la unicidad del identificador de cada agente. El CMA hace que este middleware sea compatible con el estándar FIPA, ya que implica tener un AMS en un sistema multiagente.

- *CDA*: Es el agente directorio del cluster, donde se registran todos los agentes de aplicación (y los servicios que ofrecen) que se encuentran en un cluster. El CDA hace este middleware compatible con el estándar FIPA, ya que implica tener un DF en un sistema multiagente.

Teniendo en cuenta los recursos limitados de los nodos de sensores inalámbricos, tanto el CMA como el CDA acceden al mismo repositorio de datos situado en el cabeza de cluster, eliminando la necesidad de sincronizar dos repositorios diferentes. Cada agente de aplicación sólo ofrece un servicio, por ejemplo, detección de gas, detección de fuego, etc.

#### ***b) Componentes middleware en los nodos miembros de un cluster***

- *LSDAC*: Se encuentra en todos los nodos del cluster y es el componente de adquisición de datos del sensor local, ya que se ocupa de obtener los datos de forma periódica (pudiéndose fijar la frecuencia con la que se desea obtener los datos) de los sensores locales del nodo en el que se encuentra. Estos datos se colocan en un repositorio junto con los parámetros que indican lo recientes que son los datos, estando disponibles para los agentes.

- *NMC (Node Management Component)*: Se encuentra en todos los nodos del cluster, excepto en el cabeza de cluster; y es el componente gestor de nodos, ya que se ocupa de registrar el nodo en el que se encuentra con el CMA del cluster del que forma parte. Además, se encarga de registrar con el CMA y con el CDA un agente de aplicación, cuando éste procede de un nodo miembro de un cluster vecino.

El NMC comparte con el CMA la autoridad de eliminar un agente para liberar recursos, quedando de este modo disponibles para agentes de alta prioridad que se están ejecutando en el nodo. Cuando se elimina un agente es necesario informar al CMA y al CDA del cabeza de cluster mediante un solo mensaje para que eliminen el registro del agente de aplicación del repositorio que mantienen.

- *NLM*: Se encuentra en todos los nodos del cluster y es el componente gestor de la lista de vecinos, ya que mantiene una lista actualizada de todos los nodos vecinos a un salto. Actualiza esta lista periódicamente con el fin de eliminar los vecinos que podrían haber dejado de funcionar o haber abandonado la distancia a un solo salto.

- *ACL*: Es el lenguaje de comunicación de agentes que proporciona el estándar FIPA, y mediante el cual los agentes se comunican de forma asíncrona. Para ello, cada nodo tiene una cola de mensajes (MQ - Message Queue), que contiene los mensajes para los agentes junto con sus identificadores. De este modo, cuando un mensaje llega

a una MQ de un nodo, se informa al agente correspondiente mediante el subsistema de comunicación del middleware. La comunicación puede tener lugar entre agentes del mismo nodo o entre agentes que residen en diferentes nodos. Los mensajes se pueden enviar en unicast (obteniendo anteriormente la posición del agente destino mediante el CMA) o en broadcast (cuando se quiere informar de un evento determinado a los nodos vecinos que se encuentran a un salto).

### *c) Elección de un cabeza de cluster*

Inicialmente, el cabeza de cluster se designa en el momento del despliegue de la WSN. Posteriormente, durante el funcionamiento de la WSN, cuando un cabeza de cluster tiene un nivel bajo de batería, transfiere la gestión de los agentes (transfiere el CMA y el CDA mediante el gestor de la movilidad de agentes) a uno de los nodos vecinos (que se encuentra a un salto) que tenga más nivel de batería. Si todos los nodos vecinos que se encuentran a un salto también tienen niveles bajos de batería, se busca un nodo vecino que se encuentre a dos saltos. Este proceso se repite hasta que se encuentre un nodo suplente que ejerza de cabeza de cluster, o el cabeza de cluster actual muera a causa del nivel bajo de batería. Si el cabeza de cluster muere, entonces los nodos miembros de dicho cluster sólo tienen la opción de unirse a otros clusters.

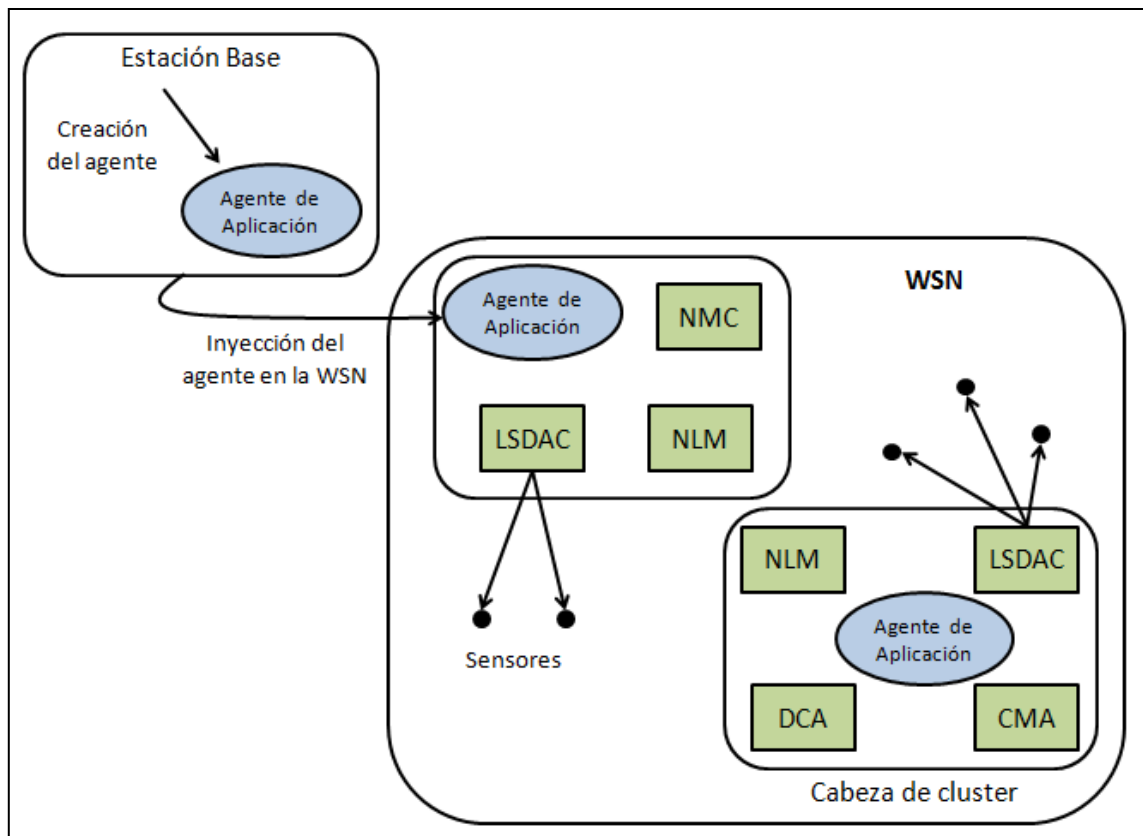


Figura 43. Modelo de funcionamiento de la arquitectura

Cabe destacar que, la estructura basada en cluster junto con los agentes de gestión de los cabezas de cluster, hace más fácil la comunicación entre agentes dentro de un mismo cluster.

## 2.6 Conclusiones

Como se ha podido observar en estos últimos apartados dónde se ha ido describiendo la tecnología de agentes, los diferentes estándares publicados para agentes móviles y estándares/propuestas para agentes en WSN, esta tecnología puede aportar grandes ventajas a la hora de implementar los servicios implantados en el HD.

En primer lugar, pensar en una solución de agentes en un HD nos lleva directamente a fijarnos en todas sus cualidades y posteriormente, ver si es lo más adecuado para todas las exigencias que someten los servicios del HD. Este primer punto, está lo suficientemente sobrepasado ya que los agentes móviles ofrecen casi todas las características indispensables como escalabilidad, movilidad, eficiencia, reducción del tráfico en las redes, robustez, inteligencia, adaptabilidad, etc.

Además, lo que van a aportar es una interoperabilidad entre dos medios de comunicación tan diferenciados como son los equipos conectados a infraestructura fija

de comunicaciones y las redes de sensores inalámbricos. Gracias a la propuesta presentada por la Universidad de Darmstadt, ha sido posible que tanto los agentes FIPA que se ejecutan en medios con recursos elevados, ya sean en forma de memoria, batería o procesamiento, puedan comunicarse con agentes de la WSN debido a que tienen un funcionamiento similar y compatible.

Por otro lado, una solución de agentes móviles en el HD va a proporcionar una solución unificada de servicios que no van a depender de múltiples tecnologías desplegadas por el hogar. Esto es así, ya que serán los agentes que se encarguen de la supervisión de los diferentes servicios y facilidades que ofrecen tecnologías como KNX o LonWorks. Por tanto, el usuario únicamente tendrá que centrar su atención en el buen funcionamiento de los diferentes agentes que se estén ejecutando en el HD.

Esta última característica es una de las más importantes, ya que normalmente en los mercados, las soluciones más sencillas para los usuarios son las más aceptadas. En este caso, proponer una solución en la que englobe multitud de servicios domóticos, y que estén controlados por una única tecnología (Agentes móviles) tiene cierto atractivo.

Por todas estas cuestiones, a continuación se presenta una propuesta de HD en la que se intenta agrupar todas las tecnologías descritas hasta ahora, controladas todas ellas por agentes móviles.

### 3 Propuesta de hogar digital

En la actualidad, la sociedad se ve envuelta en una constante búsqueda para mejorar su nivel de vida. Una de las tendencias actuales es facilitar las tareas de los usuarios en su vida cotidiana, a través de los servicios ofrecidos por el hogar digital. El enfoque que se va a realizar, pretende resolver los servicios del hogar digital con la tecnología de agentes estudiada en este TFM.

Antes de definir el modelo de agentes, es necesario hacer un pequeño análisis, para una mayor comprensión de las pasarelas existentes en el hogar digital. Se comenzará explicando desde las pasarelas más sencillas hasta llegar a la propuesta final que se va a desarrollar, para que el modelo de agentes esté coherentemente integrado en el hogar digital.

En primer lugar, se presenta la pasarela residencial, del que se explicarán sus funcionalidades principales en el hogar digital. A continuación, se expondrán dos posibles soluciones para la integración de las diferentes tecnologías coexistentes en dicho entorno. Y por último, se intentará dar una solución a la integración de las redes de sensores dentro del hogar digital.

Además, se definirán los diferentes bloques funcionales presentes, los tipos de agentes y sus funciones, así como los servicios ofrecidos al usuario en el hogar digital.

Para terminar este modelo de agentes, se describirá brevemente el funcionamiento de un servicio ofrecido en el hogar digital y la interacción entre los diferentes elementos involucrados.

#### 3.1 Objetivos

Antes de comenzar con el marco teórico de esta propuesta de HD, se expone un conjunto de objetivos que se ha intentado abordar en los siguientes subapartados:

- Integrar la tecnología de agentes móviles dentro del HD, para soportar los diferentes servicios de un HD.
- Aglutinar en un HD tanto tecnología de agentes como tecnologías no agentes (KNX, LonWorks, etc).
- Para obtener una propuesta mucho más interesante en cuanto a posibles servicios a soportar, integrar también en esta propuesta un posible escenario con redes WSN, además de las ya mencionadas en el punto anterior.

- Proporcionar de cara al usuario una solución única de uso de sus servicios dentro del HD.
- Realizar un breve estudio sobre las diferentes pasarelas existentes en los HD y su modo de funcionamiento para soportar diferentes servicios y/o aplicaciones.
- Definir una pasarela residencial que sea capaz de interoperar con diferentes tecnologías, ya sea tecnologías de agentes o tecnologías no agente, y que sea nexo de unión entre ellas.
- Definir una arquitectura de agentes dependiendo de su funcionalidad y de su especificidad. De esta manera, tendremos una pirámide estratégica, dónde cada uno de los niveles quedará definido qué funcionalidades deben soportar esos tipos de agente.
- Definir un conjunto de posibles servicios que se podrían implementar en este HD con tecnología de agentes.
- Enumerar qué elementos hardware o software estarán involucrados en esta propuesta.

### **3.2 Pasarela residencial**

El concepto de pasarela residencial se entiende como un elemento existente en el hogar digital que tiene la funcionalidad de interconectar las diferentes redes domésticas con el exterior.

En la siguiente figura se puede ver un posible escenario de actuación de la pasarela residencial, con todos los servicios capaces de ofrecer:

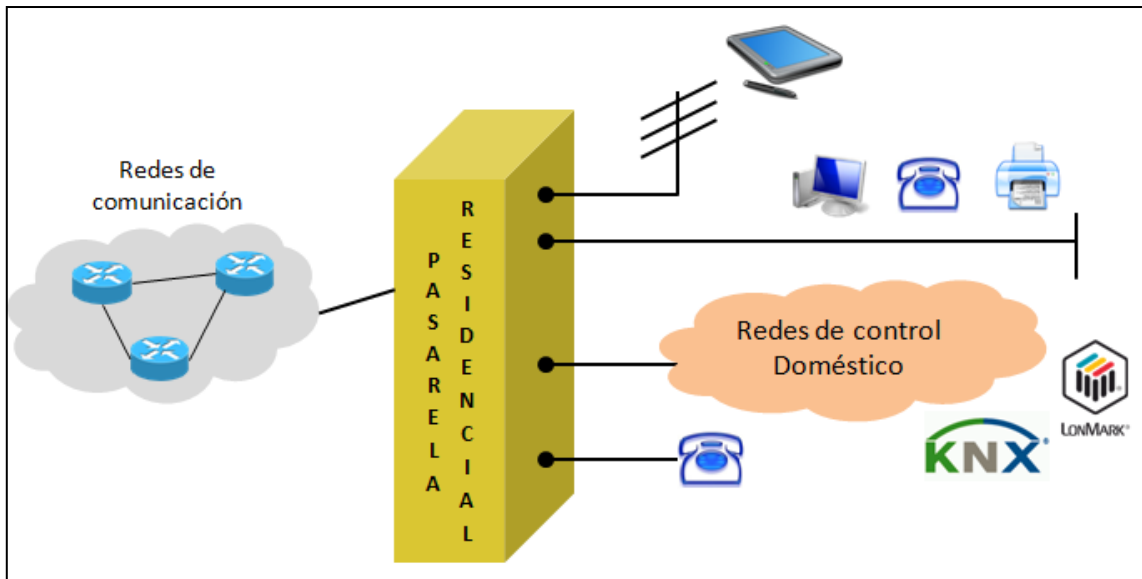


Figura 44. Adaptador Multimedia Interactivo

La pasarela residencial puede actuar como elemento de acceso a redes externas, por ejemplo, para la compra de música, películas, etc., como decodificador de televisión (Canal Plus) o distribuidor de películas por cable con acceso a Internet (televisión digital), se utiliza también como conmutador entre los diferentes dispositivos que forman parte de la red doméstica. Para ello, este elemento de pasarela residencial deberá tener funcionalidades de router, conmutador Ethernet, módem con acceso a internet, etc.

Debido a las numerosas tecnologías existentes y a la falta de interacción entre ellas, se ve necesaria la existencia de un estándar que defina un marco de trabajo en el hogar digital, a través del cual, las diferentes compañías puedan desarrollar sus servicios y aplicaciones sin tener en cuenta aspectos de bajo nivel. Por ello, empresas como Sun Microsystems, IBM, Telefónica o Alcatel, entre muchas otras han creado OSGi (Open Services Gateway Initiative) [41], un grupo de trabajo encargado de definir una arquitectura software que permita diseñar plataformas compatibles para proveer múltiples servicios (Figura 45).



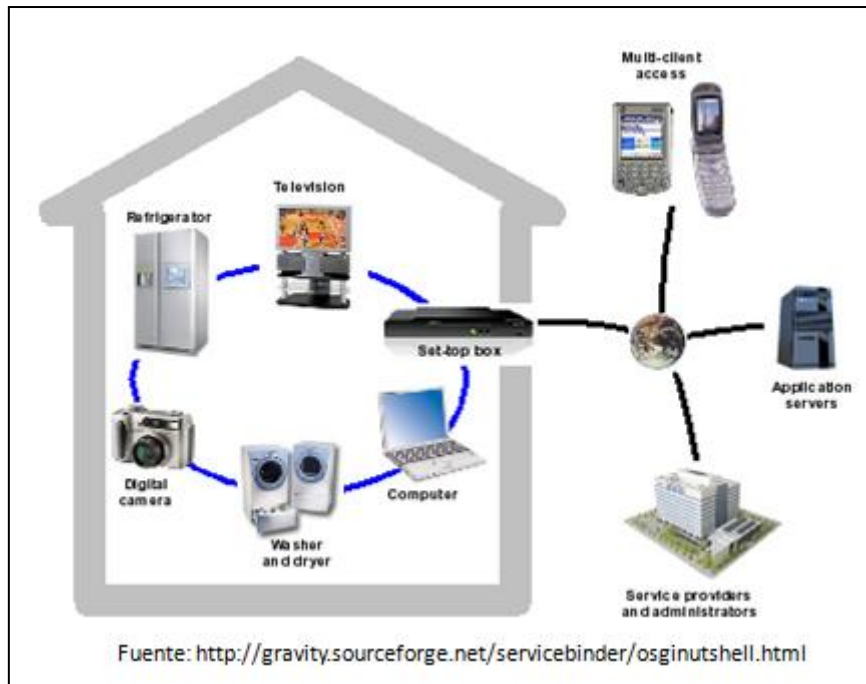


Figura 45. Marco de actuación de OSGi

Dentro de este marco de trabajo, la pasarela residencial será un servidor integrado en la red que conectará Internet (redes externas) con los distintos dispositivos internos del hogar digital, es decir, se insertará en un punto intermedio entre los proveedores de servicios y la red doméstica, tanto flujos de datos como multimedia. Además, se especifican una serie de API que define la funcionalidad de dicha pasarela, utilizando, en la medida que sea posible, los estándares existentes de Java.

Siguiendo con este enfoque de pasarela residencial, se plantea la primera solución para la integración de las diferentes tecnologías del hogar digital. Esta primera solución son un tipo de pasarelas residenciales básicas, cuyo cometido es la conmutación entre tecnologías en base a un parámetro en el tipo de mensaje.

La siguiente figura muestra una pila de protocolos que describe esta pasarela:

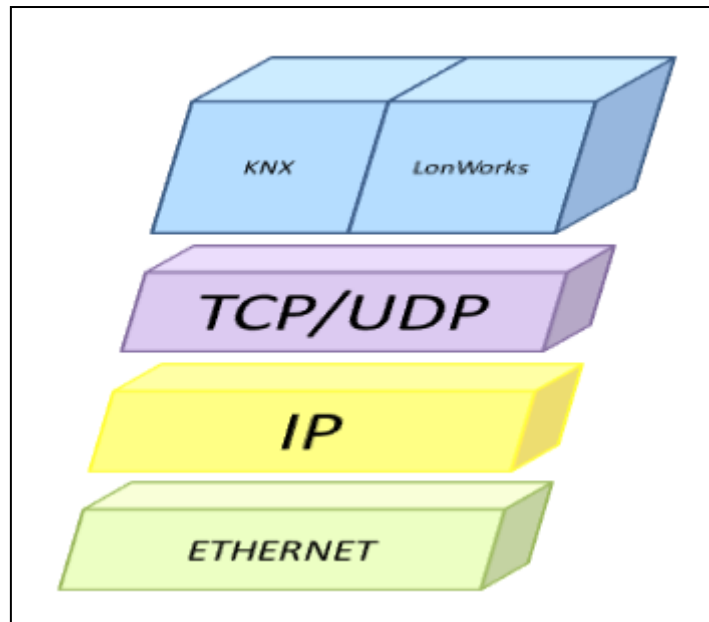


Figura 46. Pasarela básica

Como se puede observar en el dibujo, por un lado, la pasarela entenderá el lenguaje de los protocolos tradicionales como IP, y por el otro será capaz de comprender los protocolos en los que se basen cada una de las tecnologías del hogar digital, ya sean, KNX o LonWorks (que son las que se van a tratar en esta propuesta).

El funcionamiento de esta pasarela es muy básico (o “tonto”). Cada mensaje recibido desde el nivel IP, será procesado por una aplicación de nivel de aplicación que será capaz de obtener un campo discriminador (nivel de aplicación), de modo que, según ese campo del mensaje, el destino será uno u otro. Es decir, dependiendo del valor del discriminador, el mensaje a generar por esta aplicación será de tipo KNX, LonWorks o lo que corresponda, y será entregado a un dispositivo adecuado.

La propuesta se basará en este tipo de pasarelas y con la inclusión de algún agente de por medio, en esta propuesta se conseguirá conectividad entre las tecnologías anteriormente mencionadas y el estándar de agentes FIPA.

### 3.3 El libro blanco de Telefónica

Telefónica publicó este libro [42] el 3 de Julio de 2003 como una solución más a la integración de diferentes servicios que se ejecutan sobre múltiples o diferentes tecnologías de red, ya sea Wi-Fi, WiMax, UWB, etc. Además, en él se describe más que una simple solución a un problema concreto, sino que define entre otras cosas, un

estado del arte actual, modelo de negocio y un análisis a corto plazo de la evolución del hogar digital.

La novedad principal en este nuevo escenario que Telefónica nos presenta, es que en el hogar van a coexistir elementos tales como una línea de Banda Ancha (conexión al exterior), redes de datos (para la conexión de impresoras, ordenadores de mesa, escáneres...), redes multimedia (interconexión de televisiones, cadenas de radio, DVD, altavoces...), redes domóticas (sensores y actuadores) y una pasarela residencial (es el elemento que interconecta las distintas redes domésticas para poder tener acceso desde el exterior del hogar).

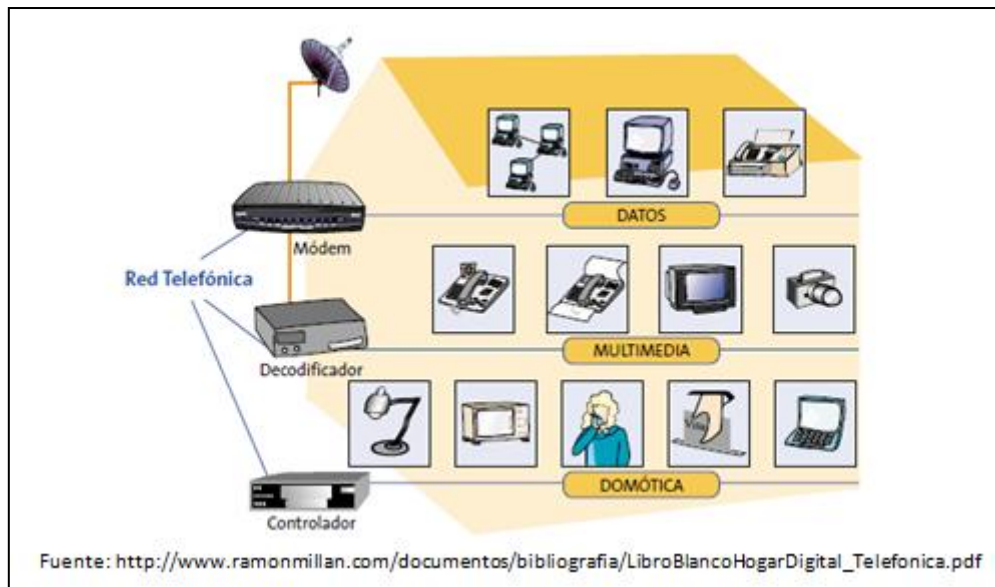


Figura 47. Escenario de actuación de Telefónica

Como se puede observar, el elemento clave en el que Telefónica se va a basar para poder manejar y tener una comunicación absoluta de todas las redes existentes en el hogar digital, es la pasarela residencial. Ésta es necesaria ya que en ese momento no existía ningún dispositivo o tecnología que ofreciese una solución completa e integrada, además de otras barreras como los sistemas propietarios o la carencia de infraestructuras adecuadas en los hogares.

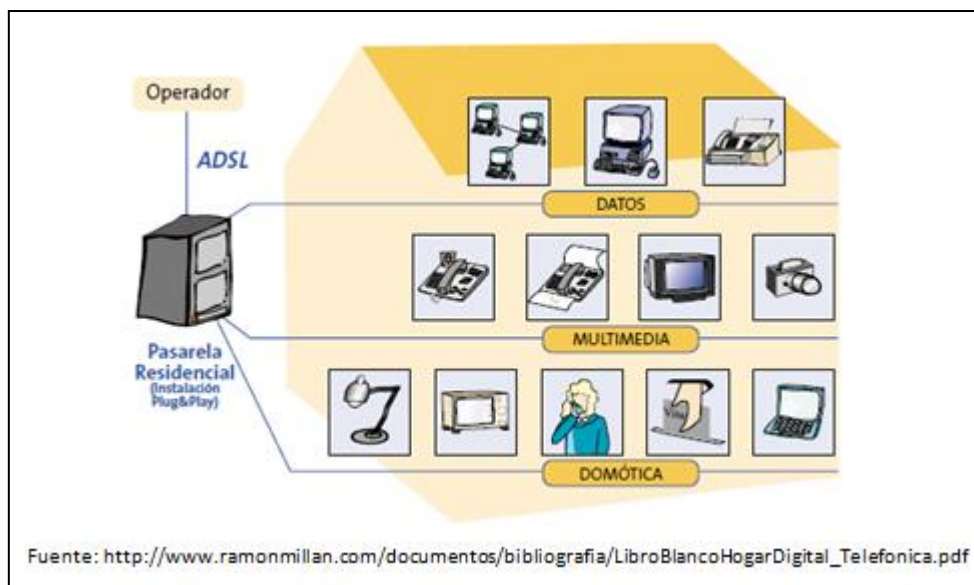


Figura 48. Integración de la pasarela en el escenario

Esta pasarela residencial, consigue ofrecer unas características como la convergencia de redes, facilidad de configuración de los dispositivos con tecnología *plug&play*, un único punto de acceso al hogar y la utilización de estándares abiertos que permiten un mayor grado de interoperabilidad entre los equipos y las aplicaciones.

Con todo esto, un usuario podrá acceder desde el exterior a los servicios que tenga habilitados en el hogar, gracias a que la pasarela residencial hace como punto de acceso único y además, porque tiene la capacidad de entender y administrar los diferentes equipos y aplicaciones que están instalados en el hogar.

Para hacer realidad este modelo de negocio o propuesta, Telefónica decide dividirla en tres partes: gestor del portal, agregador de servicios e integrador de servicios.

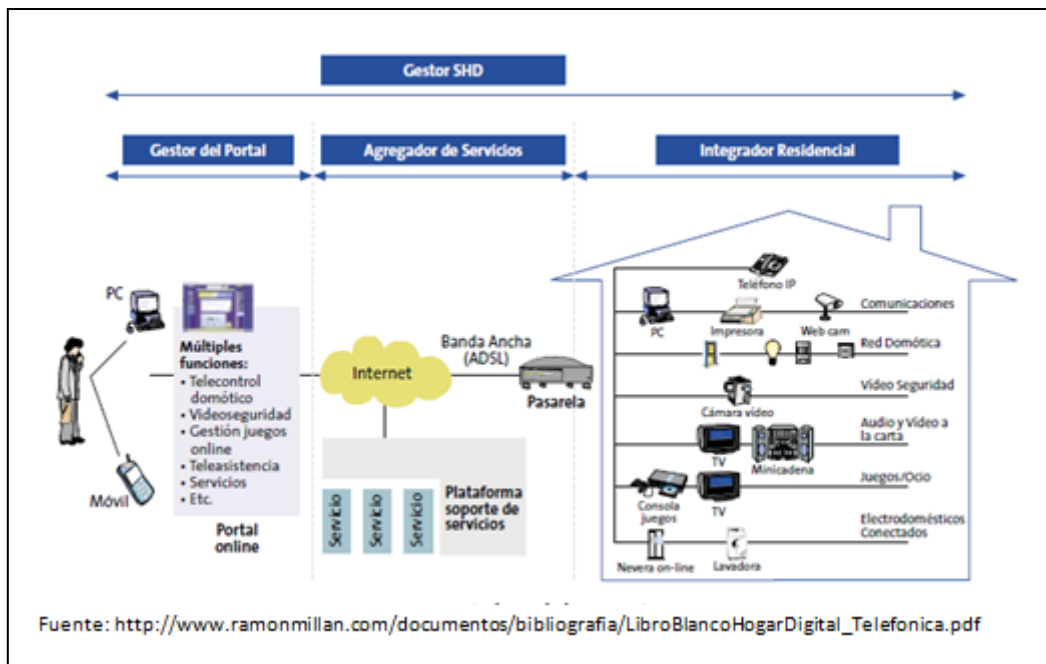


Figura 49. Partes de la pasarela de Telefónica

### 3.3.1 El gestor del portal

Los servicios proporcionados a los usuarios son accedidos y controlados a través de este portal. Por lo que, si un usuario desea modificar algún parámetro o servicio en funcionamiento dentro del hogar digital, accederá a dicho gestor por medio de Internet.

### 3.3.2 Agregador de servicios

La funcionalidad principal de este elemento es la de proporcionar los requisitos y niveles de conectividad necesarios para ofrecer y soportar los servicios que el proveedor presta al usuario. En definitiva, lo que va a hacer es garantizar la agregación de nuevos servicios solicitados por el usuario, integrarlos dentro del hogar digital, gestión de servicios y garantizar que los servicios son accesibles para los usuarios.

Todo esto, hace que la instalación de aplicaciones o servicios sea mucho más sencilla para el usuario, ya que no tiene que preocuparse de cómo montar y mantener los servicios que desea, debido a que este módulo contiene y soporta todas estas funcionalidades que muchos usuarios resultan engorrosas.

### 3.3.3 Integrador residencial

Este módulo o elemento se refiere al hogar digital. Es aquí donde los servicios se van a instalar y mantener para facilitar a los usuarios la vida cotidiana. En ella, pueden coexistir, como ya se ha comentado, diferentes tecnologías y redes, todas ellas controladas por la pasarela residencial.

Como se ha podido observar, Telefónica ha buscado un elemento capaz de soportar a la vez diferentes redes y servicios, y además, su instalación y mantenimiento para el usuario es bastante sencillo y abstracto.

El motivo por el que se hace referencia a este Libro Blanco de Telefónica, es porque uno de los objetivos de esta propuesta es la de poder soportar diferentes tecnologías y servicios, a través de un elemento pasarela con un software basado en la tecnología de Agentes que, a continuación se detallará.

### 3.4 Pasarela Residencial basada en Agentes (PRA)

A continuación se describe la propuesta de modelo de agentes que se ha desarrollado en este TFM. Este modelo de agentes se basa en el siguiente escenario dividido en tres partes:

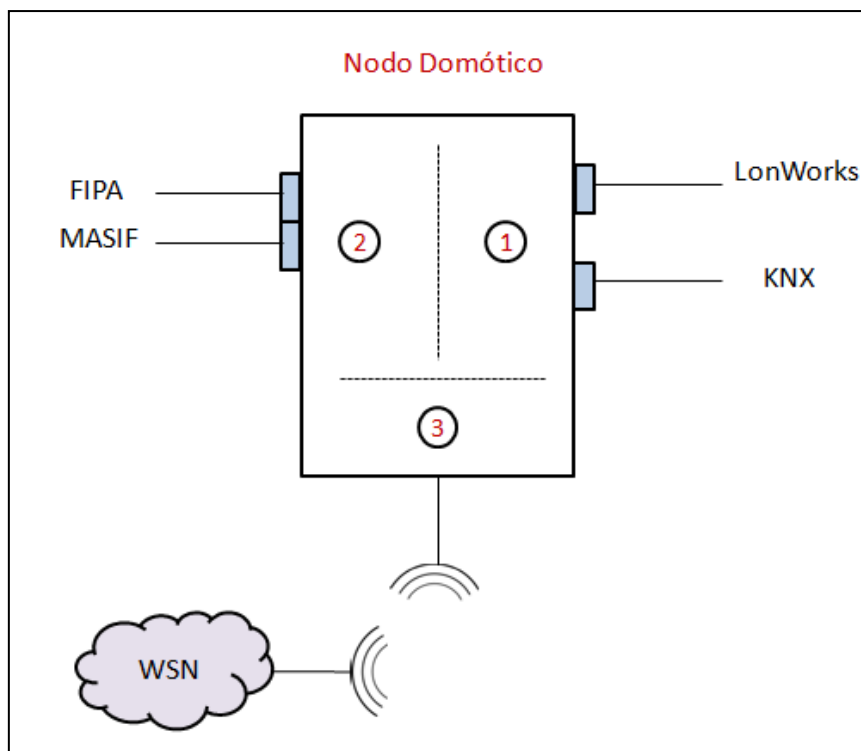


Figura 50 - Escenario de actuación del Nodo Domótico

- **Parte 1:** Interfaces con las tecnologías domóticas instaladas en el HD.
- **Parte 2:** Plataforma de agentes (nodo domótico).
- **Parte 3:** Plataforma de Agentes para WSN (Wireless Sensor Network).

El siguiente paso, va a consistir en la explicación de estas partes, de manera que cada una se vaya integrando con lo explicado anteriormente. Este proceso de explicación irá incrementando la funcionalidad de la pasarela a medida que se vaya acercando a su punto final. Una vez explicada la última parte, se tendrá una visión global y minuciosa de este modelo de agentes.

### 3.4.1 Interfaces con las tecnologías domóticas instaladas en el HD

Este primer bloque funcional de la pasarela residencial que se está desarrollando en este TFM, va a contener una funcionalidad prácticamente similar al modelo de pasarelas básicas, explicado en un apartado anterior.

Su funcionalidad principal, va a ser la de conmutar las peticiones/respuestas de las diferentes tecnologías ya existentes en el hogar digital, como KNX o LonWorks. Para ello, esta pasarela residencial tendrá diferentes entradas dependiendo de una tecnología u otra (por ejemplo, plug&play o RJ-45). Además, será necesaria la inclusión de una aplicación que sea capaz de reconocer los diferentes tipos de datos que va recibiendo por las múltiples entradas (deberá conocer qué entrada pertenece a cada tecnología) y a la vez, ser capaz de traducir dichas peticiones/respuestas a la tecnología destino.

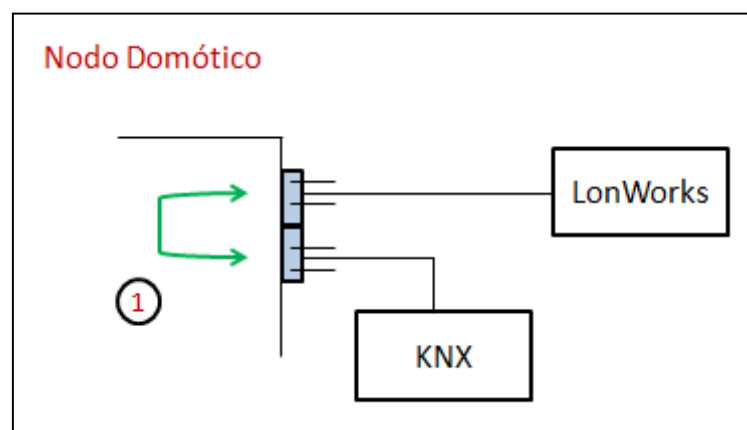


Figura 51 - Traducción de protocolos LonWorks-KNX

### 3.4.2 Plataforma de Agentes (Nodo Domótico)

El objetivo de este segundo bloque funcional es interconectar las distintas plataformas existentes en el hogar digital (KNX, LonWorks...) con las diversas plataformas que soporten los estándares de agentes (FIPA o MASIF). Por lo tanto, en el nodo domótico va existir una parte que va a integrar una plataforma de agentes sobre la que se ejecutarán los diferentes agentes y que también se podrá comunicar con agentes en otros nodos.

Para que exista una comunicación entre los diferentes agentes que se ejecutan en el nodo domótico con las diferentes tecnologías no agente ya integradas en el hogar digital (KNX o LonWorks), es necesario incluir un programa software (que perfectamente pudiera ser un agente estático) que sea capaz de convertir las peticiones de los agentes a órdenes concretas para dichas tecnologías no agente. Para ello, este programa tendrá que tener diseñado en una base de datos una lista que englobe todas las posibles peticiones de los agentes e inmediatamente ligado a estas entradas, su correspondiente orden a la tecnología concreta del hogar digital. De esta manera, existe interoperabilidad entre el mundo de los agentes y el mundo de tecnologías no agente.

La opción adoptada de introducir una plataforma de agentes en este nodo domótico, es debido a como se ha ido observando durante este TFM, la tecnología de agentes proporciona ciertas características muy interesantes, como son autonomía (no necesita la actuación de las personas humanas para funcionar correctamente), inteligencia (sabe actuar según los eventos que le van sucediendo), facilidad de comunicarse con otros agentes (es importante, ya que los diferentes servicios existentes en el HD estarán implementados por esta tecnología), etc.

La ubicación de dicho nodo podrá ser un PC que implemente las tecnologías de agentes utilizadas en el hogar o cualquier otro dispositivo electrónico con la suficiente capacidad de memoria y procesamiento necesario para un correcto uso de dichas plataformas de agentes.

Una de las características que se pueden apreciar a primera vista, es que el agente residente en él va a ser un agente estático (aunque en un futuro, tras la definición de las interfaces podría ser necesario que fuese un agente móvil), ya que no va a tener la necesidad de viajar de una tecnología a otra, debido a que va a estar situado en punto central de la comunicación y por este punto pasarán todas las solicitudes que no pertenezcan a la tecnología origen. La siguiente imagen muestra una estructura posible de conexión, aunque también podrían instalarse varios nodos domóticos específicos para cada par de tecnologías.



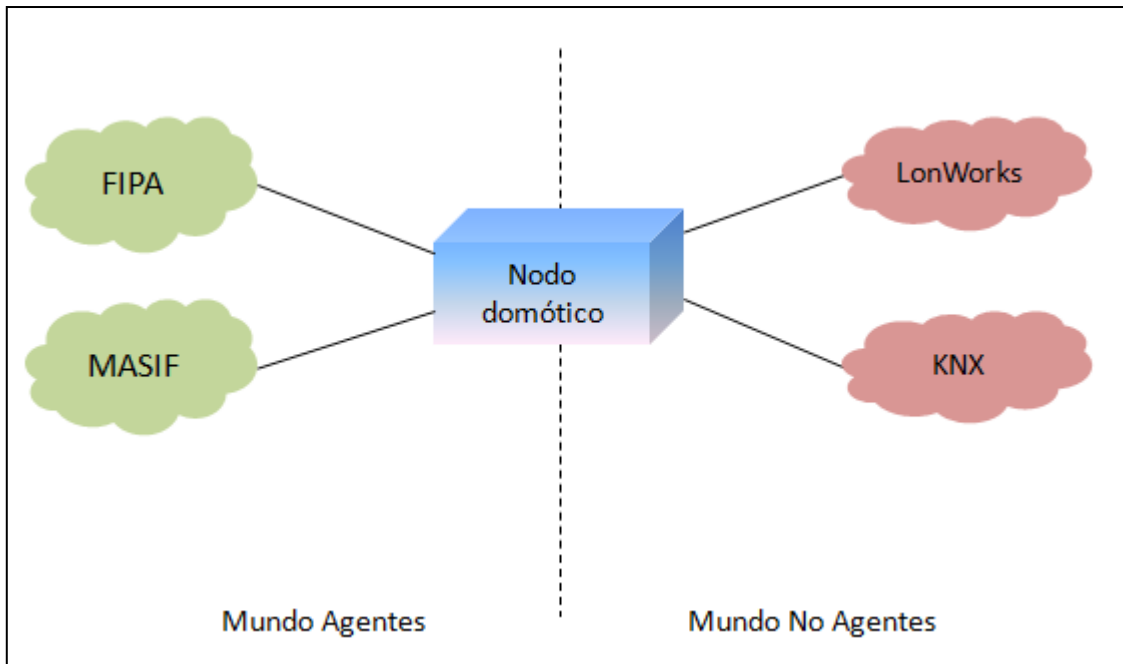


Figura 52 - Estructura genérica de conexión del nodo doméstico

Siguiendo en este ámbito de descripción genérico, vamos a tener dos pilas de protocolos a ambos lados del nodo doméstico. Cada una de estas pilas de protocolos pertenecerá a una tecnología diferente, como anteriormente se ha comentado. En la siguiente figura se puede ver un pequeño ejemplo:

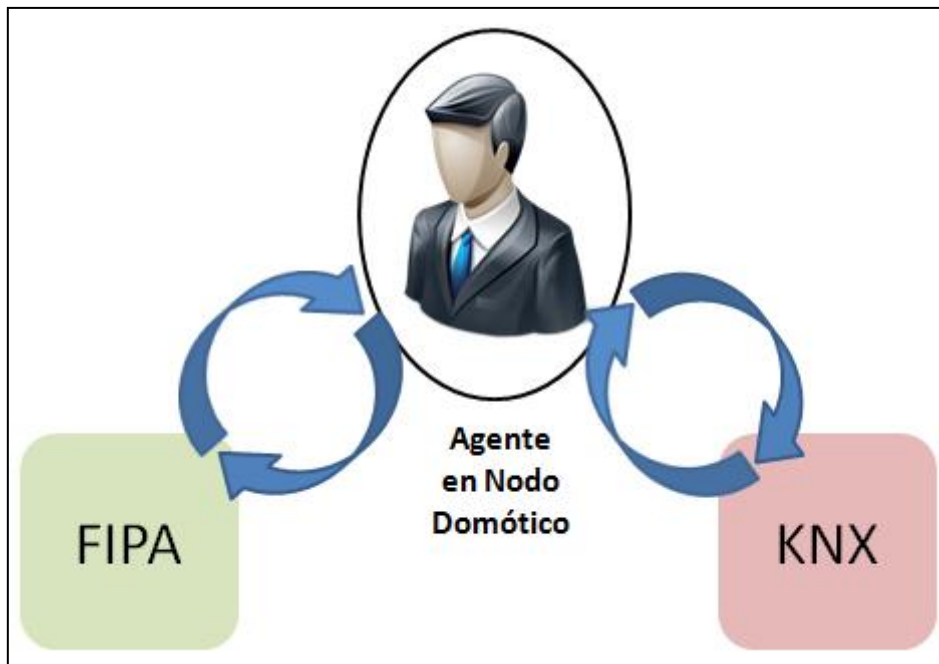


Figura 53 - Estructura específica de conexión del nodo doméstico

Como se ha podido observar, existen dos pilas de protocolos de dos tecnologías diferentes, como son FIPA y KNX. El ejemplo que vamos a detallar será el siguiente: un dispositivo instalado en FIPA debe enviar una orden a un dispositivo conectado en el bus de control de KNX para levantar una persiana. Para ello, el proceso que se lleva a cabo es el siguiente:

a) Un agente FIPA decide levantar una persiana, debido a que de acuerdo con las preferencias del usuario sabe que debe levantar esa persiana a cierta hora del día (preferencias que, por ejemplo, puede haber aprendido detectando que el usuario levanta la ventana a cierta hora todos los días, o simplemente porque el usuario lo ha definido en una interfaz de preferencias). Nótese que este es el tipo de funcionalidades que los agentes pueden aportar. Por tanto, construirá el mensaje adecuado para ello en el formato correspondiente de FIPA.

b) A continuación, usando un directorio de agentes buscará un agente que dé el servicio de interactuar con las persianas (o más en general con dispositivos domóticos). Como respuesta a la búsqueda averiguará el identificador del agente en el nodo domótico y enviará dicho mensaje.

c) Una vez la petición llegue al agente en el nodo domótico, éste agente tendrá un nuevo objetivo que es levantar la persiana. El agente sabe las acciones que puede hacer para cumplir ese objetivo, en ese caso generar un telegrama KNX para hacer la petición a la red KNX.

d) Una vez transformada la petición, es decir, tenemos el telegrama KNX completo, el nodo domótico lo transmitirá, llegando, en este caso, al bus de control de KNX.

e) Una vez en la plataforma KNX, se analizará la petición, recibéndola el dispositivo adecuado, y éste levantará la persiana.

En la siguiente figura, se puede ver el funcionamiento explicado anteriormente:

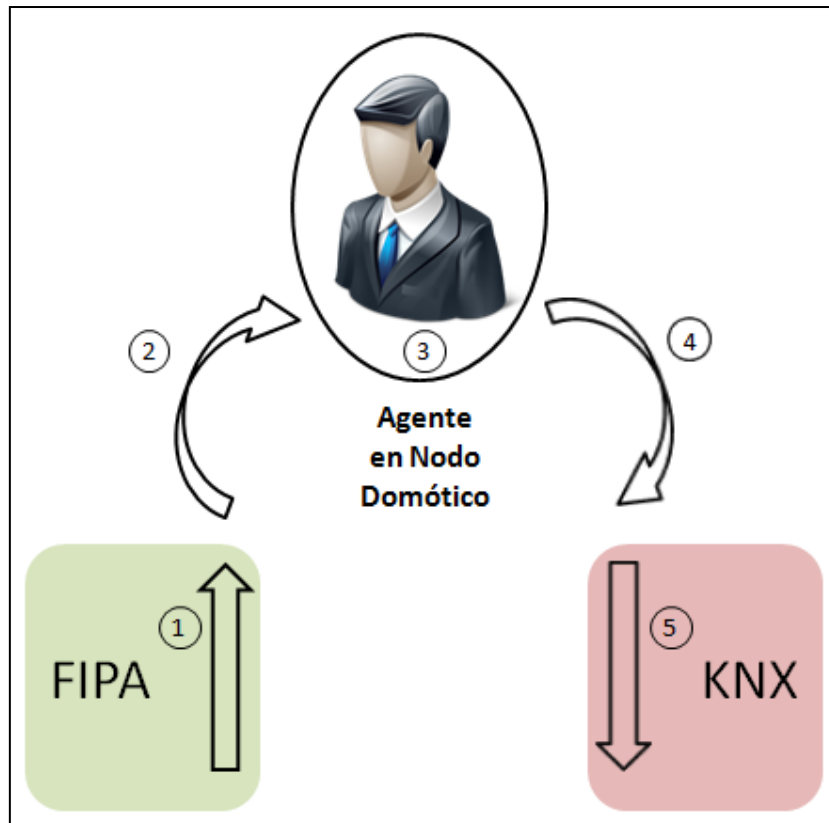


Figura 54 - Ejemplo de traducción FIPA - KNX

Todo lo expuesto hasta este momento es referente a dispositivos conectados a la red cableada, ya que tanto las tecnologías del hogar digital (LonWorks o KNX) como los estándares de agentes (MASIF y FIPA) trabajan con dichos dispositivos. La siguiente parte trata la red de dispositivos inalámbricos, es decir, la red de sensores.

### 3.4.3 Plataforma de agentes para WSN

Esta tercera parte se va a basar en una propuesta ya existente (explicada con detalle en la sección 2.5.3 de este TFM), en la que la arquitectura de la red de sensores va a ser compatible con el estándar FIPA. Por este motivo, se ha elegido esta propuesta de red de sensores ya que en este modelo de agentes está integrado el estándar FIPA.

Antes de comenzar a explicar cómo se va a integrar esta red de sensores elegida, se recordará en líneas generales la arquitectura que utiliza esta red.

Esta red de sensores que se va a desplegar por el hogar digital está dividida en clusters, donde cada uno de ellos está gestionado por un nodo cabeza de cluster. Existen dos tipos de nodos: los nodos cabeza de cluster y los nodos miembro.

Los nodos cabeza de cluster constan de dos elementos que son implementados como agentes, CMA (Cluster Management Agent) y CDA (Cluster Directory Agent), encargados de gestionar los agentes que llegan al nodo, y registrar los servicios que prestan estos agentes. Estos dos componentes son los que hacen que esta propuesta sea compatible con FIPA.

El resto de elementos que componen esta arquitectura no son modelados como agentes. Estos elementos son: el LSDAC (Local Sensor Data Acquisition Component), encargado de la adquisición de datos de los sensores; el NLM (Neighbor List Manager), que se encarga de la gestión de la lista de vecinos de un nodo; y el NMC (Node Management Component), encargado de gestionar los agentes en los nodos miembros.

Para que la comunicación entre agentes sea posible se va a utilizar el lenguaje de comunicación de agentes que proporciona el estándar FIPA. Para ello, cada nodo del cluster va a constar de una cola de mensajes (MQ - Message Queue) que contendrá los mensajes junto con un identificador del agente destino. Cuando un mensaje llegue a un nodo, se informará al agente correspondiente mediante subsistema de comunicación del middleware.

Una vez definida la arquitectura de la red de sensores, se han estudiado las posibles soluciones para integrarla en la presente propuesta. Debido a que el agente incluido en el nodo doméstico es capaz de entender el estándar FIPA, va a ser capaz de comunicarse con el CMA del nodo cabeza de cluster de la WSN.

La comunicación entre el nodo doméstico y el nodo cabeza de cluster se llevará a cabo por medio de mensajes FIPA entre el agente del nodo doméstico y el agente CMA.

Una vez explicadas todas las partes de la propuesta y teniendo completamente definido este modelo de agentes, se va a exponer un ejemplo detallado de cómo funcionaría la comunicación entre las diferentes tecnologías nombradas a lo largo de esta explicación.

#### ***Ejemplo: Automatización de ventanas***

En primer lugar, se va a definir el escenario de actuación de este ejemplo, y a continuación, se detallará el procedimiento que se llevará a cabo entre las distintas tecnologías implicadas en la comunicación.

Imagínese que en su hogar digital tiene en una habitación una planta, la cual tiene la necesidad de que se le suministre una cierta cantidad de luz en determinados momentos del día. Para ello, se ha decidido instalar a su alrededor una pequeña red de sensores (WSN) que se encargará de detectar el estado de la planta (humedad y

temperatura entre otros). En un momento cualquiera los sensores detectarán si hay luz o no, pero saber la necesidad de luz implica ya a un agente que pueda tener conocimiento de la luz que necesitan las plantas, e iniciarán un proceso mediante el cual se activará un dispositivo que elevará la persiana de la ventana.

El proceso de comunicación que se llevará a cabo implica las tecnologías de WSN, FIPA y KNX. A continuación, se describe los pasos que se completarán para llevar a cabo este servicio de automatización de persiana:

- a) El sensor detecta el nivel de luz, y acto seguido, el componente LSDAC recogerá dichos datos del sensor. Anteriormente, el agente de aplicación que recibe los datos recogidos deberá ser registrado por el NMC en el CMA, y a la vez registrará sus servicios en el CDA.

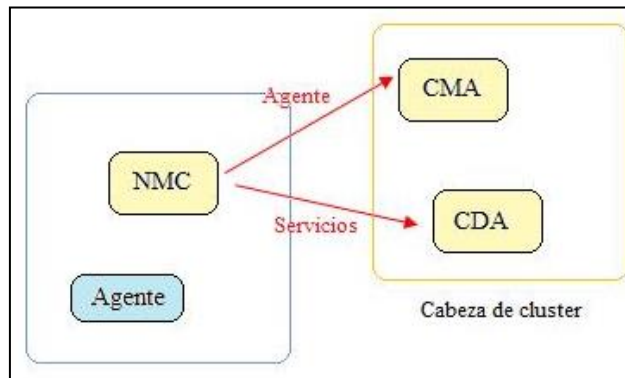


Figura 55 - Registro de un agente

- b) El agente de aplicación podrá migrar consultando la lista de vecinos del nodo en el que se encuentra, a través del componente NLM, hasta llegar al CMA del cabeza cluster.
- c) El CMA se comunica con el agente del nodo doméstico a través de mensajes FIPA para intercambiar la información deseada.
- d) Una vez los datos se encuentren en el nodo doméstico, su funcionalidad principal en este punto es la de averiguar para qué tecnología destino son los datos que acaba de recibir, transformándolos de este modo en el mensaje adecuado. Como el dispositivo encargado de levantar la persiana está controlado por KNX, se formará un telegrama KNX.
- e) Cuando el telegrama esté completamente formado, el agente lo transmitirá a la aplicación residente en el nodo doméstico, y ésta elegirá de sus múltiples

salidas por cuál de ellas debe enviar el telegrama, llegando de este modo al bus de control de KNX.

- f) Una vez en la plataforma KNX, se analizará la petición, recibiendo el dispositivo adecuado, y éste levantará la persiana.

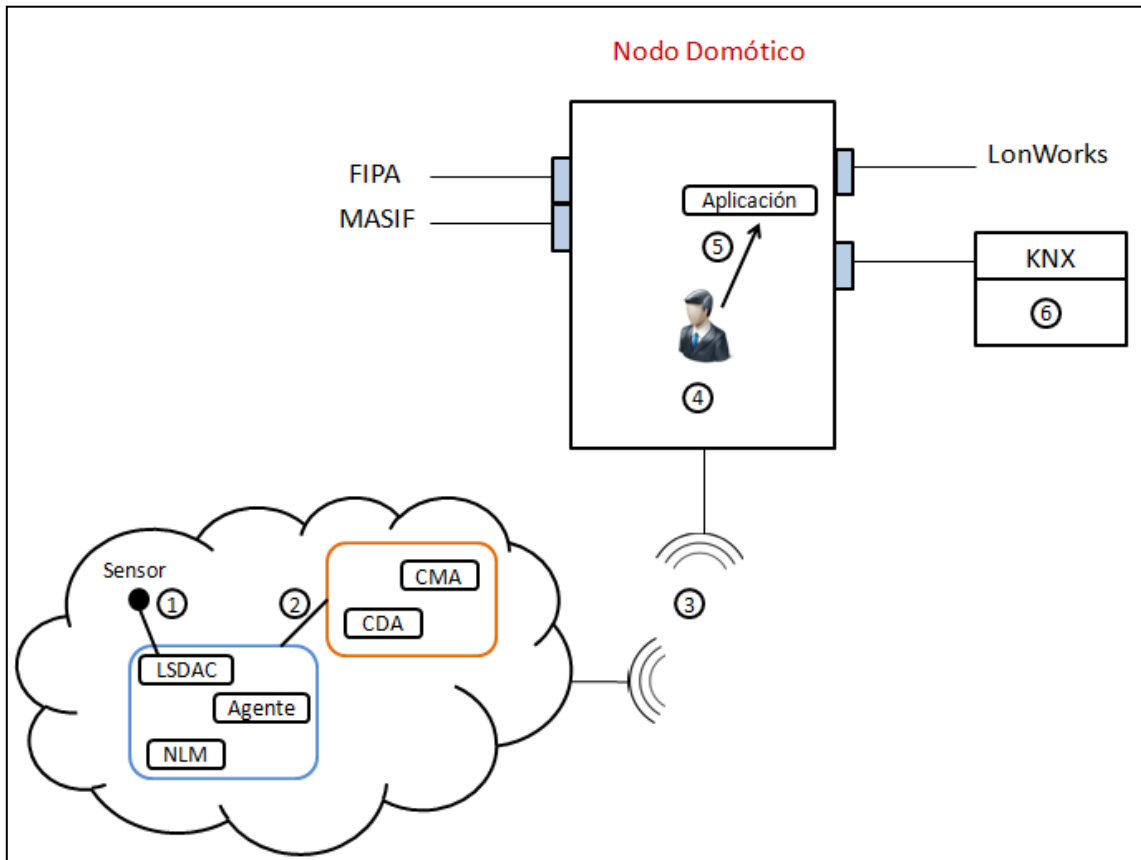


Figura 56 - Ejemplo completo del escenario de esta propuesta

### 3.5 Tipos de agentes

Antes de definir los diferentes servicios que se van a ofrecer en el hogar digital, se va a hacer un análisis de los distintos tipos de agentes con los que se va a conseguir ofrecer dichos servicios.

Dentro de la estructura de agentes que se va a implantar en el hogar digital, se han definido tres tipos de agentes atendiendo a las funcionalidades que realicen, por lo que va a existir tres rangos, que son los siguientes:

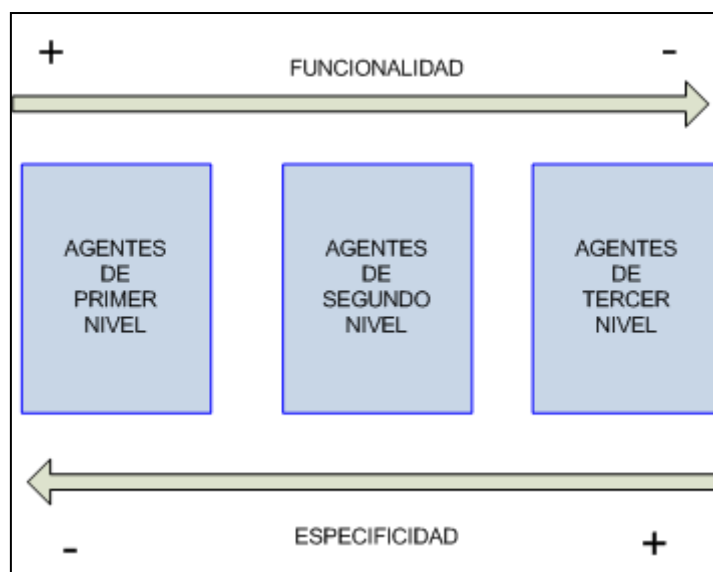


Figura 57. Tipos de agentes

### 3.5.1 Agente de primer nivel

Se va a encargar de facilitar al usuario la comprensión, adaptación y manejo del Gestor de Agentes. Dentro de este nivel, el único agente existente es el Agente Mayordomo. Este agente tiene autoridad sobre el resto de agentes de segundo y tercer nivel, es decir, podrá manejar a su antojo al resto de agentes, encargándoles determinadas tareas en los momentos que considere oportunos.

El Agente Mayordomo constará de una interfaz de usuario lo más sencilla e intuitiva posible (normalmente sería gráfica, un objetivo más ambicioso y difícil es usar una interfaz mediante lenguaje natural con toda probabilidad de restricciones), para que los usuarios finales sean capaces de comprender fácilmente los distintos servicios que se están ofreciendo en el Hogar Digital, y a la vez, sacar de ellos el máximo provecho posible. Además, este tipo de agente tiene un grado alto de autonomía frente a los usuarios, pero teniendo en cuenta los objetivos globales definidos por éste a través de la interfaz. Por ejemplo, eso significa que el agente tiene la capacidad de mostrar la temperatura actual de la casa, los eventos marcados en la agenda, las posibles incidencias ocurridas dentro de la casa, etc.

Este agente también se encargará de encargar la realización de tareas cotidianas dentro del hogar digital: limpieza, jardinería (dentro, por supuesto de las funciones disponibles en forma de sensores, actuadores y quizás, en la versión más avanzada, robots disponibles en la casa), aviso de reparaciones de averías o seguridad, entre otras.

### 3.5.2 Agentes de segundo nivel

Este tipo de agentes tienen una funcionalidad específica de menor envergadura, lo que supone que se van a encargar de tareas más concretas dentro de la casa, pero no tan generales como los agentes de primer nivel. Dentro de este grupo, se encontrarán los siguientes:

a) *Agente Climatizador*: Su función es controlar la temperatura existente ya sea en cualquier habitación de la casa o en el vehículo familiar.

Este agente recogerá las lecturas de temperatura de los sensores situados en las diferentes habitaciones, y las contrastará con unas temperaturas umbrales definidas por el usuario. Si alguno de los valores obtenidos de los sensores supera estos umbrales, el agente encenderá la calefacción o aire acondicionado. Este agente es por tanto flexible y autónomo. Su objetivo será mantener cierto rango de temperaturas, pero puede tener planes alternativos para conseguirlo. Por ejemplo, dependiendo de la temperatura del exterior, una opción alternativa para modificar la temperatura es abrir o cerrar una ventana.

Además, este agente da la posibilidad de que los usuarios puedan configurar distintas temperaturas umbrales para las diferentes habitaciones.

b) *Agente Niñera*: La función de este agente es monitorizar a los bebés. Se van a encargar de controlar los parámetros corporales del bebé, a través de los agentes de tercer nivel situados en los sensores. Cuando se detecte algún cambio importante en el entorno del bebé o en el propio bebé, este agente avisará a la persona responsable de él. Además, este agente es capaz de realizar un seguimiento del bebé hasta que llegue a la guardería, para una vez llegar allí pasar a un estado pasivo a la espera de notificaciones.

c) *Agente Jardinero*: Este agente se encarga del mantenimiento de las plantas del hogar. Se va a encargar de controlar diferentes parámetros relacionados con las plantas, como son la temperatura, grado de humedad, luminosidad, etc. En base a estos parámetros, realizará un seguimiento de ellos hasta que uno sobrepase un nivel establecido. Es entonces, cuando capturado dicho evento, actuará en consecuencia: proporcionando luminosidad y/o activando la función de riego.

d) *Agente Alarmas-Técnicas*: Su funcionalidad consiste en la detección de una cantidad importante de humo o gas dentro del hogar digital. Si alguno de estos casos ocurriera, éste avisaría a los usuarios de la casa con una alarma sonora para que la evacuen y además avisará a las casas colindantes de dicho evento, por si pudiera extenderse el incendio o la fuga de gas, para solventar problemas mayores.



e) *Agente Vecindad*: Es el agente encargado de compartir o recibir/enviar información con las casas inteligentes colindantes. Para poder compartir la información, existen dos opciones. La primera es que exista un espacio de memoria compartida por todos los hogares que forman parte del barrio inteligente, y la segunda, es que un usuario específico realice una petición de información a otro de manera directa.

f) *Agente Económico*: Se va a encargar de realizar de todas las transacciones económicas del usuario. Es decir, los pagos de facturas, las operaciones de cuentas corrientes, etc. Otra funcionalidad muy importante que aporta este agente, es que da la posibilidad de informar al usuario de los movimientos diarios de la bolsa. Si el usuario estuviese muy interesado, también podría realizar operación de compra-venta de acciones.

g) *Agente Seguridad*: La funcionalidad de este agente es la de detectar la presencia de las personas intrusas en el hogar digital, e incluso, en el barrio inteligente, previniendo así los robos en las casas o en los vehículos.

h) *Agente Alumbrado-Emergencia*: En el caso de que ocurra en el hogar digital un apagón completo o parcial, este tipo de agentes se activará y migrará a los dispositivos de emergencia, provocando el encendido de las luces de emergencia repartidas por las diferentes partes de la casa.

### 3.5.3 Agentes tercer nivel

Este tipo de agentes van a estar situados en los sensores que están repartidos por todo el hogar, y van a tener una funcionalidad muy limitada, básicamente detectarán y recogerán datos del entorno que les envuelve. Estos datos recogidos podrán ser datos del ambiente (temperatura, luminosidad, humedad, gases, ruido, etc.) o datos de los usuarios (parámetros corporales, constantes vitales, humor, estado de ánimo, etc.). Como se puede apreciar, son los agentes más básicos de toda la estructura de agentes desplegada en el hogar digital.

En la siguiente figura se puede ver un esquema global de todos los agente presentes en la arquitectura:

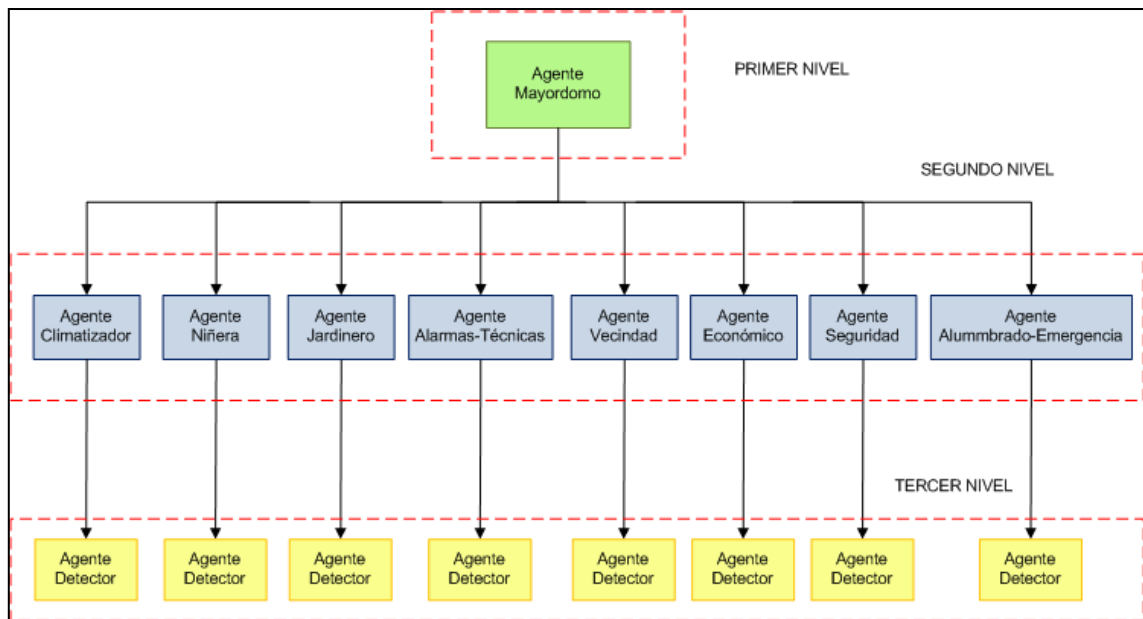


Figura 58. Árbol jerárquico de agentes

### 3.6 Servicios prestados en el HD

Sobre la infraestructura definida en los apartados anteriores planteamos en este apartado la definición de servicios:

a) *Servicio Mayordomo*: Este servicio simula al tradicional mayordomo encargado de todas las tareas cotidianas de la casa, y de los recados, ordenes o deseos del usuario. Este servicio abarca un gran abanico de funciones, que va desde el mantenimiento y control de la casa (jardinería, seguridad, limpieza, aviso de reparación de averías, etc.) hasta las necesidades del usuario: aviso de tareas programadas (citas, cumpleaños, despertador, revisiones médicas, etc.), agenda personal (añadir, modificar y eliminar contactos de la agenda), aviso de existencia de correo, entre otras muchas funcionalidades que se puedan imaginar. El agente es sobre todo un agente de interfaz, que recibe o aprende deseos de su usuario, y se comunica con otros agentes para solicitarles que lleven a cabo el cumplimiento de los objetivos correspondientes

b) *Jardinería*: El servicio de jardinería se ocupa principalmente de dos funcionalidades: mantenimiento de plantas exteriores e interiores. Este servicio se va encargar de detectar las necesidades de las plantas, como la falta de luminosidad, escasez de agua, temperaturas altas o bajas, etc. Para ello, va a ser necesario un sistema de agentes gobernado por el Agente Jardinero, el cual, tendrá bajo su tutela agentes más básicos encargados de funcionalidades muy específicas. El Agente Jardinero recogerá de forma continua las lecturas de estos agente básicos y en base a ello, llevará a cabo una determinada acción, como activar o desactivar el riego, avisar al

cortacésped domótico que debe cortar las plantas, posicionar las persianas para dejar entrar o no una determinada cantidad de luz, activar o desactivar calefactores que moderan la temperatura de las plantas, etc.

c) *Vigilancia de bebés*: Este servicio sustituirá el tradicional walkie-talkie de comunicación con los bebés, que ofrece una vigilancia de audio y video. El Agente Niñer@, encargado de este servicio, es capaz de detectar cualquier necesidad del bebé, ya sea hambre, sueño, higiene, temperatura y cualquier parámetro corporal o signo vital del bebé. Para ello, cada habitación del hogar digital tendrá instalado un conjunto de sensores, principalmente cámaras de audio y video, y detectores de signos vitales o parámetros corporales. A través de los eventos detectados, el Agente Niñer@ avisará a la persona encargada del niño. Además, este servicio tiene la capacidad de realizar un seguimiento continuo del bebé a través de los diferentes habitáculos del hogar digital, e incluso en el vehículo de transporte familiar y/o guardería. En el caso que el bebé se encuentre en la guardería, el Agente Niñer@ permanecerá en estado de pasivo a la espera de notificaciones de la guardería: hora de recogida del bebé, malestar, necesidad de ropa de recambio o pañales, urgencias médicas, etc, y además, temas relacionados con la educación o tareas realizadas por el bebé durante su estancia.

d) *Creación de ambientes*: Cuando hablamos de este servicio nos referimos a adecuar el entorno en base al estado anímico de la persona. El Agente Mayordomo será el encargado de configurar este servicio. En primer lugar, se deberá detectar la presencia del usuario y, a continuación, analiza los parámetros biométricos de la persona para asegurarse de que se trata del sueño de la casa. Una vez determinado esto, se analizarán otros factores relacionados con el estado de ánimo del usuario (expresión facial, humor, cansancio, si existe compañía o no, entre otras opciones) y en base a ello, el Agente Mayordomo, adaptará el entorno: determinada cantidad de luz, estilo y volumen la de música, le ofrece una alternativa gastronómica, le prepara el baño o no, etc.

Un ejemplo práctico de este servicio sería el siguiente: supongamos que nuestro usuario llega a casa después de un largo día de trabajo, su jefe se ha disgustado con su trabajo, por lo que el usuario se encuentra de mal humor, con un sentimiento de cansancio e impotencia por su trabajo rechazado. En este punto, el Agente Mayordomo comienza su trabajo y le prepara un ambiente con luces tenues, música relajante y un baño caliente.

e) *Seguridad en el hogar (alarmas técnicas)*: Este servicio pondrá en marcha determinados mecanismos cuando detecte peligros en el hogar del tipo de detección de incendios o fugas de gas. Cuando se detecte alguno de estos dos sucesos, el agente correspondiente activará un sistema de alarma (auditivo) para indicar a los usuarios

que evacuen el lugar. A continuación, notificará a la estación de bomberos más cercana el suceso y pondrá en funcionamiento los aspersores para solventar en lo posible el fuego o los mecanismos necesarios para minimizar la fuga de gas. Además, se informará a las casas colindantes sobre este hecho.

f) *Gestor económico*: Este servicio se va encargar de comunicarse con las diferentes entidades bancarias para realizar los pagos mensuales de facturas, actualizar las cuentas de ahorros, realizara un seguimiento diario de la bolsa y, si lo desea el usuario, tendrá la capacidad de llevar a cabo compra-venta de acciones.

g) *Climatización de la casa*: Con este servicio se activa el aire acondicionado si hace calor o la calefacción en el caso de que haga frío. Existirá un Agente Climatizador encargado de este servicio. Este agente es capaz de moverse por todas las habitaciones del hogar digital recogiendo los valores de temperatura actuales, y con estos valores y en base a una determinada temperatura umbral definida por el usuario, aumentará o disminuirá los grados existentes, encendiendo o apagando la calefacción o aire acondicionado.

h) *Teléfono hablado*: Con este servicio sustuiremos el actual teléfono fijo que tenemos en nuestras casas. Únicamente con el sonido de nuestra voz, le indicaremos al Agente Mayordomo nuestro deseo de realizar una llamada. El agente mantendrá una agenda con los números de teléfono del usuario, de modo que el usuario solo tenga que indicar el nombre del llamante y no su número. Con esto el agente iniciará el proceso de llamada.

i) *Alumbrado de emergencia*: Este servicio estará soportado por el Agente Alumbrado-Emergencia, que inicialmente se encontrará en un estado inactivo, activándose cuando se produzca un apagón en el hogar digital. Este agente se clonará y migrará a todos los dispositivos de luces de emergencia. También será aplicable no solo a un apagón en el hogar, sino también a un apagón en todo el barrio inteligente.

j) *Vigilancia del hogar o barrio inteligente*: Este servicio proporciona seguridad contra robos en el hogar o en los vehículos, en todo el barrio. Cuando un sensor detecte la presencia de un extraño en una casa o vehículo, además de activar la alarma en la propia casa y avisar a la policía, también transferirá un agente que se encargará de avisar al resto de hogares del barrio. Este agente se clonará, y será enviado a la casa vecina, donde comunicará el evento.

k) *Compartición de información entre casas inteligentes*: Con este servicio podremos intercambiar información con nuestros vecinos, como por ejemplo, archivos de música, recetas, película, libros electrónicos, etc. Para ofrecer este servicio se plantearon dos posibilidades, que son las siguientes:

- Memoria compartida por todos los hogares.
- Solicitud de los datos deseados.

Mediante la primera opción, se planteaba un espacio en el que cada dueño de una casa depositaba la información que quería compartir, y todos podían acceder a ella con total libertad, lo que ocurre con esto, es que todos los usuarios podrían acceder a la información aunque no estuvieran autorizados, pudiendo existir suplantaciones de identidad. Por este motivo se planteó la segunda solución. Cuando un usuario desea datos de otro, un agente es enviado al asistente de la casa destino solicitándole los datos, y el receptor le enviará otro agente con los datos solicitados en el caso de que decida compartirlos. El envío de los agentes entre los diferentes hogares será posible a través de una red wifi compartida.

### 3.7 Software y Hardware

Una vez definido cómo va a ser el modelo de agentes y los posibles servicios a soportar en un HD con agentes móviles como principal tecnología, es el momento de definir qué elementos hardware y software se utilizarán dentro de esta propuesta.

En cuanto a los elementos hardware que estarán involucrados para soportar todos los servicios descritos en el subapartado anterior, estarán divididos en cinco grupos.

El primero de ellos, recogerá todo tipo de equipos informáticos con amplia memoria, gran procesador, baterías de larga duración, etc. En estos equipos como por ejemplo, los equipos de sobremesa y portátiles, se estarán ejecutando/trasladando los agentes de primer y segundo nivel. Esto es debido, a que como sus características funcionales son elevadas (deben ocuparse de funcionalidades complicadas) necesitan de muchos recursos para ejecutarse. Para trasladarse entre unos equipos y otros, podrán utilizar tanto los medios cableados que pudiera poseer el HD, como medios inalámbricos, ya sea por ejemplo WiFi.

El segundo elemento hardware que albergará el HD, serán la pasarelas PRA. Estas pasarelas PRA, como se comentó en su debido subapartado, van a significar nexos de unión entre las diferentes tecnologías de agentes y tecnologías de no agentes: FIPA, WSN y KNX/LonWorks.

Otro de los elementos importantísimos en esta propuesta serán los sensores. Dichos sensores se van a encontrar soportados por la propuesta WSN de la Universidad de Damstadt y que se comunicará con el resto de tecnologías a través de las pasarelas PRA.

Complementario a los sensores, no hay que olvidar a los dispositivos que se comportan como actuadores. Estos actuadores se activarán a través de las tecnologías de KNX o LonWorks, siempre y cuando, el agente encargado de dicho servicio se lo indique a la pasarela PRA.

El último grupo hardware que alojará el HD, serán dispositivos varios con funcionalidades dispares. Este grupo consiste en un agregado de equipos hardware, que sin ellos ninguna comunicación sería capaz de realizarse. Entre ellos se puede destacar router de acceso a internet, puntos de acceso WiFi, switches, firewalls de seguridad, cableado trenzado, bases de datos para el almacenamiento de información, etc.

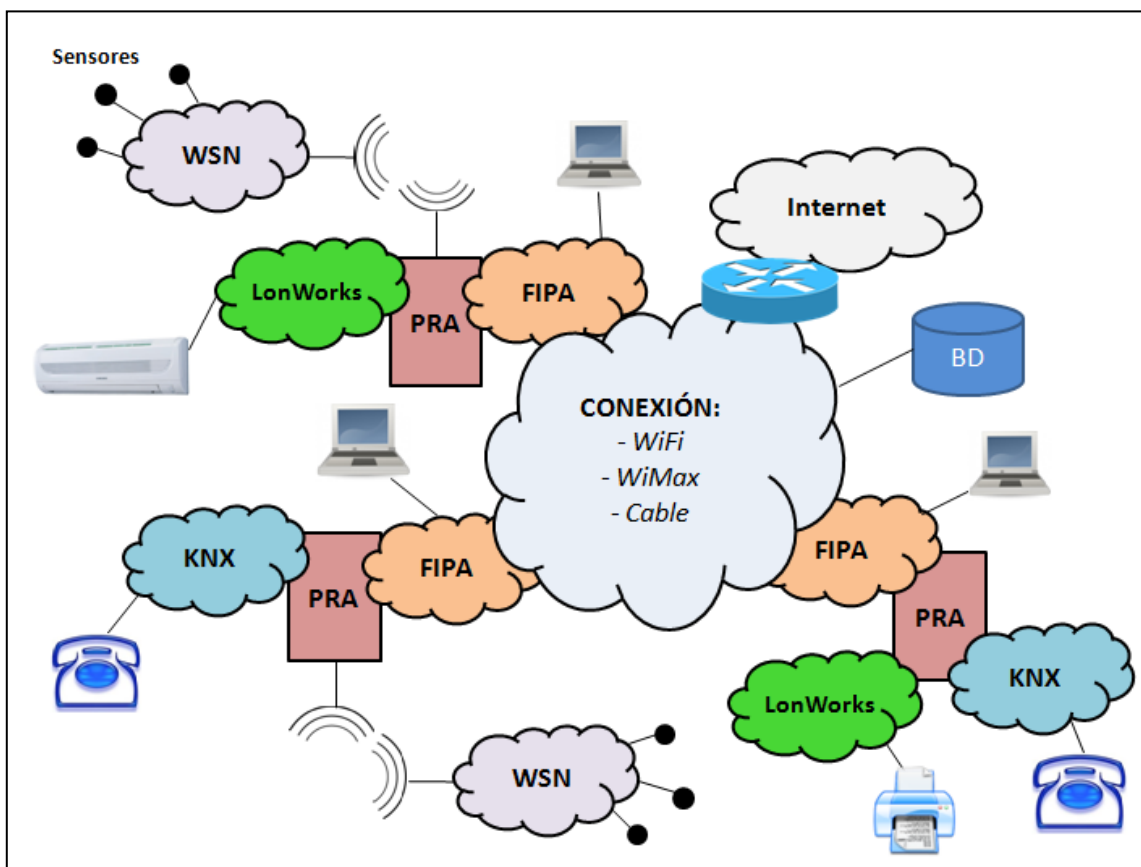


Figura 59 - Escenario hardware y software

En cuanto a los posibles elementos software presentes en el HD, son fácilmente detectables los siguientes:

- Programa que aloja la pasarela PRA. Este programa (podría ser perfectamente un agente estático) se encarga de traducir las diferentes peticiones recibidas por los agentes a órdenes entendidas por las tecnologías no agente.

- También existirá una interfaz de usuario en el que se mostrará qué servicios están activos en qué estado se encuentran cada uno de ellos. En esta interfaz de usuario (agente mayordomo), es dónde un usuario podrá interactuar con la “tecnología de agentes” para programar a su manera los diferentes servicios soportados en el HD.
- Será necesario incluir un directorio de servicio que indique qué agentes se encarga de cada uno de los servicios instalados en el HD.
- Programas de interacción con las Bases de Datos, que contienen el estado de cada uno de los sensores, de la información que contiene el HD, configuraciones almacenadas por los usuarios, etc.
- Además de todos estos programas, no hay que olvidar, que cada uno de los servicios que se describieron en el punto 3.6, deberán implementar sus correspondientes agentes, así como los diferentes mensajes a intercambiar entre ellos, las peticiones que tienen que enviar a la pasarela PRA, cómo deben funcionar, etc.

### **3.8 Funcionamiento de un servicio concreto**

Imaginemos que se quiere implementar un servicio concreto para esta propuesta, como por ejemplo, un servicio de climatización de la casa. Para ello, se va a poner en juego todo lo explicado en esta propuesta, desde los diferentes niveles de agentes, las tecnologías no agentes (KNX, LonWorks, etc), la pasarela PRA, etc.

Todo comienza con un servicio mayordomo (agente de primer nivel de la jerarquía de agentes) que está permanentemente vigilando el bienestar de los habitantes como el correcto funcionamiento de todos y cada uno de los servicios que se están ejecutando en el HD. Para este caso concreto, este servicio mayordomo se va a encargar de vigilar temporalmente la temperatura que hay en una habitación concreta del hogar. Este servicio de mayordomo, va a estar implementado a través de un agente FIPA que se ejecutará y se trasladará entre diferentes equipos conectados con un alto rendimiento y memoria.



Figura 60 - Agente Mayordomo

Por tanto, el primer paso que se va a producir en este servicio de climatización, es la creación de una petición por parte del servicio mayordomo, solicitando la temperatura actual al agente de segundo nivel que se encarga de este servicio en concreto. Este agente climatizador, al recibir la petición del servicio mayordomo, se desplazará hasta la pasarela PRA. Allí, creará un agente detector que posteriormente será inyectado en la red WSN.

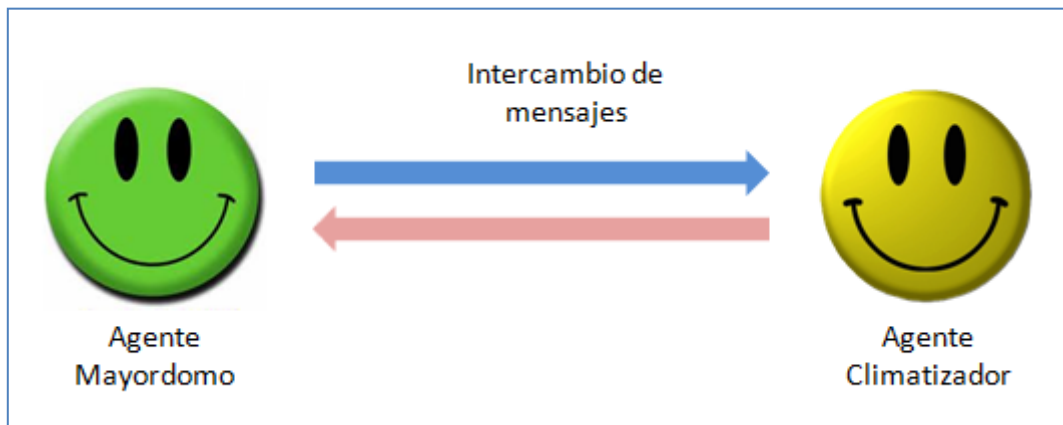


Figura 61 - Intercambio de mensajes entre agentes



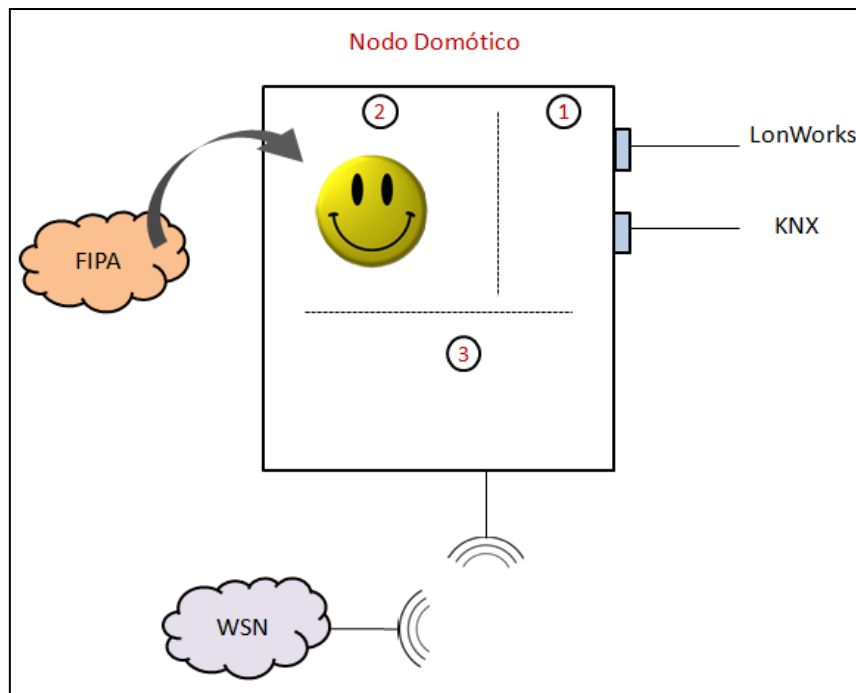


Figura 62 - Agente climatización en PRA

El funcionamiento de este agente detector dentro de la WSN, ha sido explicado ya en dos ocasiones, y más concretamente en esta propuesta en el punto 3.3.3. Por lo tanto, obviando lo que hace internamente este agente dentro de la WSN, su objetivo final será el de recoger el dato de temperatura actual que una mota está monitorizando.

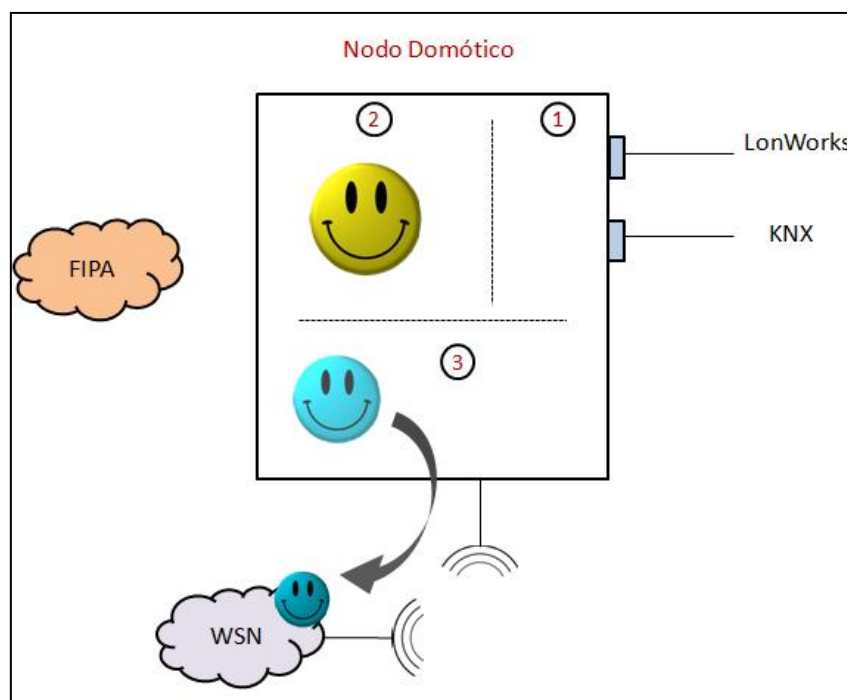


Figura 63 - Agente detector trasladándose a WSN

Una vez obtenido el dato de temperatura, este agente regresará a la pasarela PRA de la red WSN y aportará el dato al agente climatizador. Este agente, estudiará dicho dato contrastando con la configuración establecida por los usuarios del hogar. Si en el caso de que dicho dato esta fuera del rango establecido, este agente climatizador realizará una petición destinada a las tecnologías que se encargan de activar o desactivar los elementos eléctricos de la casa (en este caso, el aire acondicionado). Una vez creada esta petición de activación del aire acondicionado, el programa situado en la pasarela PRA se encargará de transformar la petición del agente climatizador a órdenes específicas a la tecnología no agentes, que ya puede ser KNX, LonWorks, etc. Dicha orden llegará al aire acondicionado, que hará accionarse automáticamente sin la interacción de los usuarios.

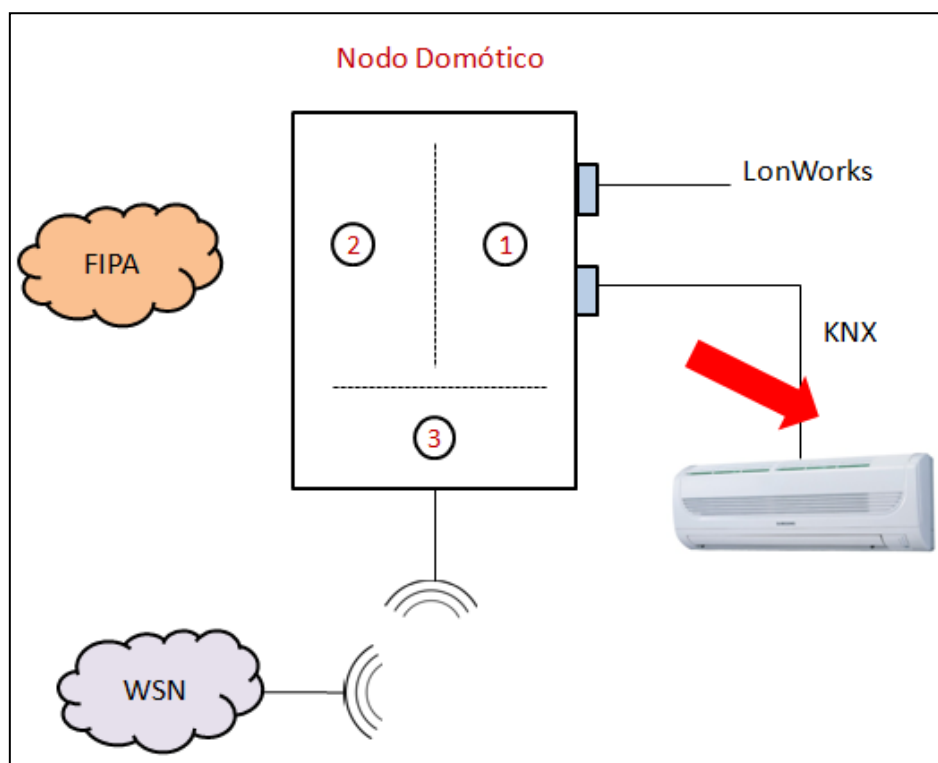


Figura 64 - Orden KNX a aire acondicionado

## 4 Conclusiones y trabajos futuros

Una de las motivaciones para la realización de este proyecto nace de la primera experiencia que tuve en Telefónica I+D con una aplicación destinada al Hogar Digital. A partir de ella, mi interés sobre este tipo de temas de investigación ha ido aumentando hasta el punto que decidí optar por él para la realización de este TFM. Además, otra de las motivaciones por las que he decidido realizar este TFM, ha sido el estar en contacto con un tema novedoso y expuesto a cambios constantes, como es el HD. Dicho tema, como cualquier persona puede suponer, posee un amplio abanico de temas bajo investigación, ya que está en pleno proceso de desarrollo e instalándose en el mercado tecnológico. En un futuro no muy lejano, los servicios ofrecidos en un HD serán mucho más potentes y llamativos que los actuales.

Otro aspecto motivador de este TFM ha sido el carácter investigador, que me ha ayudado a poder solventar los problemas que han ido apareciendo sobre la marcha. Además, al ser un tema tan innovador, dicha investigación me ha ayudado en mi continuo aprendizaje como investigador.

Para poder llevar a cabo este proyecto, ha sido necesario realizar un gran esfuerzo de búsqueda de información, ya que la tecnología de agentes, aunque no es demasiado novedosa porque nació hace varios años, no es muy prolija en cuanto a la documentación existente.

El desarrollo y las características de este trabajo han dejado abierta la posibilidad de mejorar o ampliar la investigación llevada a cabo. Se enumeran a continuación una serie de ellas, entre las que se incluyen mejoras para un futurible estudio para la realización del doctorado.

En primer lugar, sería importante realizar una pequeña aplicación FIPA en la que se pueda comprobar desde cerca, las ventajas, facilidades y comportamiento de la tecnología de agentes. Para ello, habría que realizar una breve búsqueda de “frameworks” de agentes y elegir uno de ellos. Una vez seleccionado un framework en concreto para la realización de esta prueba, se realizaría una pequeña aplicación en la que habría que luchar con las peculiaridades de dicha tecnología para poder conseguir los objetivos preestablecidos.

Una vez conocida la tecnología de agentes a bajo nivel, es el momento adecuado para implementar el elemento clave de la propuesta realizada, es decir, la pasarela de agentes. Este punto sería clave en un futurible doctorado, ya que es el elemento clave de la propuesta y debido a que en la actualidad no existe nada similar desarrollado para el HD. Por tanto, la carga de trabajo e innovación sería elevada.

Con todos los elementos anteriores conseguidos, el siguiente paso sería la adquisición de sensores para incluir la red WSN en este escenario. Para ello, se distribuiría un conjunto de motas en un espacio de pruebas que simulase un HD (con elementos LonWorks, KNX, DLNA, etc), y que implementaría la arquitectura para WSN basada en FIPA.

Para finalizar, habría que probar el funcionamiento conjunto de todos los elementos introducidos en el HD que se exponen en la propuesta. A partir de ello, se podría realizar comparaciones con otras soluciones implementadas en el HD, como por ejemplo, rendimiento, velocidades, calidad, etc.

Como conclusión general de este proyecto, agradezco haber trabajado en un tema desconocido para mí, como es la tecnología de agentes, ya que de esta manera he tenido que realizar un trabajo más personal y laborioso. Esto me ha ayudado a completar mi formación como ingeniero e investigador, por supuesto, a conocer más a fondo un mundo que me llamaba mucho la atención, como es el hogar digital. Por lo que, me siento gratamente satisfecho del trabajo realizado durante estos meses.

## Bibliografía

- [1] KNX. (2011, Marzo) KNX Association. [Online]. <http://www.knx.org/es/>
- [2] KNX. (2007, Julio) KNX ya es Estándar Internacional:. [Online]. <http://www.knx.org/fileadmin/news/120524715414725PR20061201-ES.pdf>
- [3] ISO/IEC. (2007, Junio) ISO/IEC 14543-3-6:2007. [Online]. [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=43842](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43842)
- [4] Echelon. [Online]. <http://www.echelon.com>
- [5] LonMark International. [Online]. <http://www.lonmark.org/>
- [6] LonWorld Expo. [Online]. <http://www.lonworldexpo.com/>
- [7] ZigBee Alliance. [Online]. <http://www.zigbee.org/>
- [8] FCC (Federal Communications Commission). [Online]. <http://www.fcc.gov/>
- [9] WiMedia Alliance. [Online]. <http://www.wimedia.org/en/index.asp>
- [10] (2008, Septiembre) Ventajas y Desventajas. [Online]. <http://lenguajeweb.wordpress.com/2008/09/28/ventajas-y-desventajas/>
- [11] (1997) Intelligent software agents on the internet. [Online]. [http://ebipol.p.lodz.pl/Content/1081/issues/issue2\\_3/ch\\_123/index.html](http://ebipol.p.lodz.pl/Content/1081/issues/issue2_3/ch_123/index.html)
- [12] Benjamín Vargas-Quesada Pedro Hípola. (1999, Abril) Agentes inteligentes: definicion y tipologia. Los agentes de informacion. [Online]. [http://www.elprofesionaldelainformacion.com/contenidos/1999/abril/agentes\\_inteligentes\\_definicion\\_y\\_tipologia\\_los\\_agentes\\_de\\_informacion.html](http://www.elprofesionaldelainformacion.com/contenidos/1999/abril/agentes_inteligentes_definicion_y_tipologia_los_agentes_de_informacion.html)
- [13] Joachim Baumann, Markus Straßer Fritz Hohl. Beyond Java: Merging Corba-based Mobile Agents and WWW. [Online]. [http://www.w3.org/OOP/9606\\_Workshop/submissions/20-pospaper3.html](http://www.w3.org/OOP/9606_Workshop/submissions/20-pospaper3.html)

- [14] Object Management Group. [Online]. <http://www.omg.org/>
- [15] IBM. Aglets. [Online]. <http://www.tr1.ibm.com/aglets/>
- [16] Jesus Arturo Perez Díaz. (2000, Abril) Tesis Agentes Móviles. [Online]. [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/AgentesMoviles\\_Arturo.pdf](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/AgentesMoviles_Arturo.pdf)
- [17] Hyacinth S. Nwana. (1996, Noviembre) Software Agents: An Overview. [Online]. [http://reference.kfupm.edu.sa/content/s/o/software\\_agents\\_an\\_overview\\_83303.pdf](http://reference.kfupm.edu.sa/content/s/o/software_agents_an_overview_83303.pdf)
- [18] Voyager. ObjectSpace Voyager ORB 3.1.1. [Online]. <http://www.inf.fu-berlin.de/lehre/WS99/VS/Misc/Voyager/API/>
- [19] Aglets. Welcome to Aglets Workbench! [Online]. <http://www.mip.sdu.dk/~ikk/IA/awb/aglets/readme.html>
- [20] Università di Bologna. SOMA: Secure and Open Mobile Agent. [Online]. <http://lia.deis.unibo.it/Research/SOMA/>
- [21] University of Kaiserslautern. The Ara Platform for Mobile Agents. [Online]. [http://www.wagss.informatik.uni-kl.de/Projekte/Ara/index\\_e.html](http://www.wagss.informatik.uni-kl.de/Projekte/Ara/index_e.html)
- [22] Mitsubishi Electric News Bureau. MITSUBISHI ELECTRIC INFORMATION TECHNOLOGY CENTER AMERICA. [Online]. <http://www.merl.com/projects/concordia/WWW/Concordia1.1-Release.htm>
- [23] H. Robert Frost. Documentation for the Java(tm) Agent Template, Version 0.3. [Online]. <http://www-cdr.stanford.edu/ABE/documentation/>
- [24] Stanford University. JATLite Introductory FAQ. [Online]. <http://www-cdr.stanford.edu/ProcessLink/papers/JATL.html>
- [25] J.Yaguez. Capítulo 8. Arquitecturas y Middleware de Comunicaciones. [Online]. <http://halley.ls.fi.upm.es/~jyaguez/pdfs/Agentes.pdf>
- [26] Oracle. Conozca más sobre la tecnología Java. [Online]. <http://www.java.com/es/about/>
- [27] ISO. TC1/SC22/WG21 - The C++ Standards Committee. [Online]. <http://www.open-std.org/jtc1/sc22/wg21/>

- [28] TCL. Welcome to the Tcl Developer Xchange! [Online]. <http://www.tcl.tk/>
- [29] OMG. MASIF - Update of Revised MAF Submission. [Online]. [La comunicación entre agentes está fuera del ámbito de MASIF, y es ampliamente dirigida por CORBA como comunicación de objetos.](#)
- [30] FIPA. Welcome to FIPA! [Online]. <http://fipa.org/>
- [31] Portal forestal. (2005, Junio) Una red de sensores toma el pulso a un bosque. [Online]. <http://www.portalforestal.com/component/content/21.html?task=view&sectionid=4>
- [32] Chien-Liang Fok. (2008, Abril) A Mobile Agent Middleware for Wireless Sensor Networks. [Online]. <http://mobilab.cse.wustl.edu//projects/agilla/>
- [33] Chien-Liang Fok, G.-C. Roman, and Chenyang Lu. (2005, Junio) Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications. [Online]. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?isnumber=30952&arnumber=1437126&count=77&index=61](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=30952&arnumber=1437126&count=77&index=61)
- [34] C.-L. Fok, G.-C. Roman, and C. Lu. (2005, Abril) Mobile Agent Middleware for Sensor Networks: An Application Case Study. [Online]. [http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=1440953&queryText%3DMobile+Agent+Middleware+for+Sensor+Networks%3A+An+Application+Case+Study%26openedRefinements%3D\\*%26searchField%3DSearch+All](http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=1440953&queryText%3DMobile+Agent+Middleware+for+Sensor+Networks%3A+An+Application+Case+Study%26openedRefinements%3D*%26searchField%3DSearch+All)
- [35] D. Georgoulas and K. Blow. (2008, Junio) Intelligent Mobile Agent Middleware for Wireless Sensor Networks: A Real Time Application Case Study. [Online]. [http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4545510&queryText%3D.QT.Intelligent+Mobile+Agent+Middleware+for+Wireless+Sensor+Networks%3A+A+Real+Time+Application+Case+Study%26openedRefinements%3D\\*%26searchField%3DSearch+All](http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4545510&queryText%3D.QT.Intelligent+Mobile+Agent+Middleware+for+Wireless+Sensor+Networks%3A+A+Real+Time+Application+Case+Study%26openedRefinements%3D*%26searchField%3DSearch+All)
- [36] TinyOS. (2011) MICA2. [Online]. <http://www.tinyos.net/scoop/special/hardware#mica2>
- [37] Chris Allick, John Charnas, Alex Hickman, Damon Tyman Jens Mache.

- (2009) Sensor Network Lab Exercises Using TinyOS and MicaZ Motes. [Online]. <http://www.google.es/url?sa=t&source=web&cd=1&ved=0CBkQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.72.5666%26rep%3Drep1%26type%3Dpdf&rct=j&q=MicaZ%20%20Sensor%20Network%20Lab%20Exercises%20Using%20TinyOS%20and%20MicaZ%2>
- [38] Jie Zhang Dan Weinberg. (2008) Tmote Sky. [Online]. <http://faculty.uml.edu/ylyuo/Teaching/AdvCompArch/reports/WeinbergZhangFinalReport.pdf>
- [39] TinyOS. TinyOS. [Online]. <http://www.tinyos.net/>
- [40] K. Nawaz, P.E. Guerrero, and A.P. Buchmann. (2008, Mayo) Towards a FIPA compliant multiagent based middleware architecture for sensor networks. [Online]. [http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4556226&queryText%3DTowards+a+FIPA+Compliant+Multiagent+based+Middleware+Architecture+for+Sensor+Networks%26openedRefinements%3D\\*%26searchField%3DSearch+All](http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4556226&queryText%3DTowards+a+FIPA+Compliant+Multiagent+based+Middleware+Architecture+for+Sensor+Networks%26openedRefinements%3D*%26searchField%3DSearch+All)
- [41] OSGi Alliance. OSGi Alliance Community Event 2011. [Online]. <http://www.osgi.org/Main/HomePage>
- [42] Telefónica. (2003, Julio) Libro Blanco del Hogar Digital y las Infraestructuras Comunes de Telecomunicaciones. [Online]. [http://www.ramonmillan.com/documentos/bibliografia/LibroBlancoHogarDigital\\_Telefonica.pdf](http://www.ramonmillan.com/documentos/bibliografia/LibroBlancoHogarDigital_Telefonica.pdf)
- [43] (2007, Febrero) [Online]. <http://halley.ls.fi.upm.es/~jyaguez/pdfs/cortomiddlewarefebrero2007.pdf>
- [44] J. Postel. (1980, Agosto) [Online]. <http://www.ietf.org/rfc/rfc0768.txt>
- [45] Alvaro Ortigosa. (2008, Abril) Programación Orientada a Objetos - Aplicaciones Distribuidas. [Online]. <http://arantxa.ii.uam.es/~ortigosa/poo/08rmi.pdf>
- [46] Sun Microsystems. (1995, Agosto) RPC: Remote Procedure Call Protocol Specification Version 2. [Online]. <http://tools.ietf.org/html/rfc1831>
- [47] Sun Microsystems. Remote Method Invocation Home. [Online].



- <http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>
- [48] Object Management Group. (2008, Noviembre) [Online].  
<http://www.omg.org/docs/formal/08-11-06.pdf>
- [49] 1.- Introducción. [Online].  
<http://www.terra.es/personal2/monkiki/doc/CORBA/intro.html>
- [50] Network Working Group J. Boyer. (2001, Marzo) RFC3076 - Canonical XML Version 1.0. [Online]. <http://www.faqs.org/ftp/rfc/pdf/rfc3076.txt.pdf>
- [51] Network Working Group E. O'Tuathail. (2002, Junio) RFC3288 - Using the Simple Object Access Protocol (SOAP). [Online].  
<http://www.faqs.org/ftp/rfc/pdf/rfc3288.txt.pdf>
- [52] OASIS. (2004, Octubre) UDDI Version 3.0.2. [Online].  
<http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>
- [53] W3C. (199, Diciembre) HTML 4.01 Specification. [Online].  
<http://www.w3.org/TR/html4/>
- [54] W3C. (2007, Julio) WSDL 1.1 Element Identifiers. [Online].  
<http://www.w3.org/TR/wsdl11elementidentifiers/>
- [55] ZigBee Alliance. ZigBee Alliance. [Online]. <http://www.zigbee.org/>
- [56] Echelon. Echelon Corporation, embedded control networking technology. [Online]. <http://www.echelon.com/>
- [57] LonMark International. LonMark International. [Online].  
<http://www.lonmark.org/>
- [58] LonWorld. (2007, Marzo) LonWorld 2007. [Online].  
<http://www.lonworldexpo.com/>
- [59] (2008, Septiembre) Ventajas y Desventajas - Lenguaje Web. [Online].  
<http://lenguajeweb.wordpress.com/2008/09/28/ventajas-y-desventajas/>
- [60] Intelligent Software Agents on the Internet. [Online].  
[http://ebipol.p.lodz.pl/Content/1081/issues/issue2\\_3/ch\\_123/index.html](http://ebipol.p.lodz.pl/Content/1081/issues/issue2_3/ch_123/index.html)
- [61] Benjamín Vargas-Quesada Pedro Hípola. (1999, Abril) Agentes inteligentes:

- definicion y tipologia. Los agentes de informacion. [Online].  
[http://www.elprofesionaldelainformacion.com/contenidos/1999/abril/agentes\\_inteligentes\\_definicion\\_y\\_tipologia\\_los\\_agentes\\_de\\_informacion.html](http://www.elprofesionaldelainformacion.com/contenidos/1999/abril/agentes_inteligentes_definicion_y_tipologia_los_agentes_de_informacion.html)
- [62] Joachim Baumann, Markus Straßer Fritz Hohl. (1996, Abril) Beyond Java: Merging Corba-based Mobile Agents and WWW. [Online].  
[http://www.w3.org/OOP/9606\\_Workshop/submissions/20-pospaper3.html](http://www.w3.org/OOP/9606_Workshop/submissions/20-pospaper3.html)
- [63] OMG. Object Management Group. [Online]. <http://www.omg.org/>
- [64] IBM. (2002, Marzo) Aglets. [Online]. <http://www.trl.ibm.com/aglets/>
- [65] Hyacinth S. Nwana. (1996, Noviembre) Software Agents: An Overview. [Online].  
[http://reference.kfupm.edu.sa/content/s/o/software\\_agents\\_an\\_overview\\_83303.pdf](http://reference.kfupm.edu.sa/content/s/o/software_agents_an_overview_83303.pdf)
- [66] ObjectSpace Voyager. (1999) ObjectSpace Voyager 3.1. [Online].  
<http://www.inf.fu-berlin.de/lehre/WS99/VS/Misc/Voyager/API/>
- [67] IBM. IBM Aglets Workbench. [Online].  
<http://www.mip.sdu.dk/~ikk/IA/awb/aglets/readme.html>
- [68] Università di Bologna. SOMA: Secure and Open Mobile Agent. [Online].  
<http://lia.deis.unibo.it/Research/SOMA/>
- [69] Dartmouth College. (2000, Septiembre) D´Agents: Mobile Agents at Dartmouth College. [Online]. <http://agent.cs.dartmouth.edu/>
- [70] University of Kaiserslautern. (1997, Octubre) The Ara Platform for Mobile Agents. [Online].  
[http://www.wagss.informatik.uni-kl.de/Projekte/Ara/index\\_e.html](http://www.wagss.informatik.uni-kl.de/Projekte/Ara/index_e.html)
- [71] Mitsubishi Electric Information Technology Center America. (1998, Junio) MITSUBISHI ELECTRIC INFORMATION TECHNOLOGY CENTER AMERICA UPGRADES CONCORDIA MOBILE AGENT FRAMEWORK. [Online].  
<http://www.merl.com/projects/concordia/WWW/Concordia1.1-Release.htm>
- [72] H. Robert Frost. (1996) Java(tm) Agent Template, Version 0.3. [Online].  
<http://www-cdr.stanford.edu/ABE/documentation/>

- [73] Stanford University. (1996) JATLite Description. [Online]. <http://www-cdr.stanford.edu/ProcessLink/papers/JATL.html>
- [74] Stefan Covaci. (1998, Septiembre) The OMG Mobile Agent System Interoperability Facility. [Online]. <http://209.85.229.132/search?q=cache:9ct82X2Sdzg:www.objs.com/isig/inter-net-98-09-12.ppt+enables+interoperability+between+agent+platforms+of+different+vendors&cd=1&hl=es&ct=clnk&gl=es>
- [75] Object Management Group, Inc. (2000, Mayo) OMG Document. [Online]. <http://www.omg.org/cgi-bin/doc?formal/2000-06-40>
- [76] Object Management Group, Inc. (2000, Enero) Mobile Agent Facility. [Online]. [http://www.omg.org/technology/documents/formal/mobile\\_agent\\_facility.htm](http://www.omg.org/technology/documents/formal/mobile_agent_facility.htm)
- [77] J.Yaguez. 8.6 Modelos orientados a agentes móviles. [Online]. <http://halley.ls.fi.upm.es/~jyaguez/pdfs/Agentes.pdf>
- [78] Sun Microsystems. Conozca más sobre la tecnología JAVA. [Online]. <http://www.java.com/es/about/>
- [79] ISO/IEC. (2009, Mayo) ISO/IEC JTC1/SC22/WG21. [Online]. <http://www.open-std.org/jtc1/sc22/wg21/>
- [80] Tcl Developer Xchange. Tcl Developer Site. [Online]. <http://www.tcl.tk/>
- [81] UMBC. UMBC AgentWeb. [Online]. <http://www.cs.umbc.edu/kqml/>
- [82] IEEE. Welcome to the Foundation for Intelligent Physical Agents. [Online]. <http://www.fipa.org/>
- [83] Juan Pavón Mestras. (2005) Agentes Móviles. [Online]. <http://www.fdi.ucm.es/profesor/jpavon/doctorado/AgentesMoviles.pdf>
- [84] T. Magedanz C. Bäumer, *The Grasshopper Mobile Agent Platform Enabling Shortterm Active Broadband Intelligent Network Implementation*, Sahin Albayrak, Ed. Berlín, Alemania: Springer, 1999.

- [85] Christoph Bäumer, Markus Breugst, Sang Choy Thomas Magedanz. Grasshopper - A Universal Agent Platform Based on OMG MASIF and FIPA Standards. [Online]. <http://cordis.europa.eu/infowin/acts/analysys/products/thematic/agents/ch4/ch4.htm>
- [86] Antonio Corradi, Cesare Stefanelli Paolo Bellavista. (1999) IEEE Xplore - Login. [Online]. <http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F6754%2F18058%2F00838439.pdf%3Farnumber%3D838439&authDecision=-203>
- [87] Guenter Karjoth Mitsuru Oshima. (1997, Mayo) Aglets Specification. [Online]. <http://www.trl.ibm.com/aglets/spec10.htm>
- [88] IBM. (2009, Febrero) SourceForge.net: Aglet Software Development Kit. [Online]. <http://sourceforge.net/projects/aglets/>
- [89] Sun. Java SE Downloads - Sun Developer Network. [Online]. <http://java.sun.com/javase/downloads/index.jsp>
- [90] Diatel UPM. (2003, Julio) CasaFutura: la domótica desde un punto de vista telemático. [Online]. <http://casafutura.diatel.upm.es/>
- [91] Inmaculada Díaz, MODELO DE AGENTES EN EL HOGAR DIGITAL SOBRE UNA PLATAFORMA HETEROGÉNEA, Julio 2009, Pendiente de presentación.
- [92] Pablo E. Guerrero, and Alejandro P. Buchmann Khalid Nawaz, "Towards a FIPA Compliant Multiagent based Middleware Architecture for Sensor Networks," *IEEE Xplore*, pp. 339-343, Julio 2008.
- [93] María Dolores Jimenez, MODELO DE AGENTES EN EL HOGAR DIGITAL SOBRE UNA PLATAFORMA HETEROGÉNEA, Junio 2009, Presentado.
- [94] Jesús Arturo Pérez Díaz. (2000, Abril) TesisAgentesMóviles. [Online]. [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/AgentesMoviles\\_Arturo.pdf](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/AgentesMoviles_Arturo.pdf)