

Universidad Carlos III de Madrid



Escuela Politécnica Superior

Ingeniería Informática

Proyecto Fin de Carrera

Computación Ubicua en la Universidad: Agentes

MUSSUS: Multi-agent Ubiquitous System to Supply
University Services

Autor: Erika Moreno Sierra

Tutor: Javier Carbó Rubiera

Mayo 2009

A mi familia, que me ha soportado a mí y a mi ordenador por medio todo este tiempo y siempre me ha apoyado en todo, incluso cuando no veían que este proyecto pudiera tener fin.

A mis amigos, especialmente a Ana y Jaime, que son la prueba de que existen amigos para siempre y no solo para los buenos momentos.

Y a Diego, por apostar por mí y estar siempre a mi lado y por hacer todo lo posible por encontrar ese camino que seguro nos llevará muy lejos.

Índice

1. INTRODUCCIÓN.....	5
2. ESTADO DEL ARTE	10
COMPARATIVA Y CONCLUSIONES.....	21
3. OBJETIVOS Y PLANTEAMIENTO DEL PROYECTO MUSSUS	24
¿QUÉ ES MUSSUS?.....	24
AGENTES DE LA APLICACIÓN	25
<i>Agentes usuarios</i>	25
<i>Agentes servidores</i>	25
TECNOLOGÍAS UTILIZADAS.....	27
ESCENARIO DE USO	28
4. ROLES, AGENTES Y ONTOLOGÍA DE MUSSUS.....	29
MODELO DE ROLES.....	29
<i>Análisis de recursos</i>	29
<i>Análisis de interfaces externas</i>	31
<i>Análisis de roles</i>	32
<i>Especificación de roles</i>	33
MODELO DE AGENTES.....	42
REPRESENTACIÓN GRÁFICA DE CONCEPTOS.....	43
<i>Agente Secretaría</i>	43
<i>Agente Biblioteca</i>	44
<i>Agente Cafetería/Comedor</i>	44
<i>Agente Centro Cultural/Deportivo</i>	45
5. PROTOCOLOS DE MUSSUS.....	46
PROTOCOLOS GENERALES	46
<i>Movimiento de usuarios</i>	46
<i>Envío de información</i>	49
<i>Desconexión de usuarios</i>	51
<i>Desconexión de zonas</i>	53
<i>Desconexión del sistema</i>	54
PROTOCOLOS DEL AGENTE SECRETARÍA.....	55
<i>Notificación de entregas en el casillero (sólo para profesores)</i>	55
<i>Notificación de reclamaciones (sólo para profesores)</i>	56
<i>Notificación de publicación de notas / Posibilidad de escribir una reclamación a una nota (sólo para alumnos)</i>	57
<i>Recordatorio de plazos de entrega de trabajos/prácticas (sólo para alumnos)</i>	58
<i>Recordatorio de fechas y lugares de exámenes (sólo para alumnos)</i>	59
<i>Notificación de localización de un profesor al que se está buscando</i>	60
<i>Notificación de localización de un alumno al que se está buscando</i>	61
<i>Notificación de que un objeto perdido ha sido encontrado</i>	62

PROTOCOLOS DEL AGENTE BIBLIOTECA	64
<i>Recordatorio de plazos de entrega de ejemplares (sólo para alumnos)</i>	64
<i>Recordatorio de días sancionados, en caso de que existan (sólo para alumnos)</i> ..	65
<i>Servicio de reserva de aulas de estudio (sólo para alumnos)</i>	66
<i>Recordatorio de aulas reservadas (sólo para alumnos)</i>	67
<i>Recordatorio de libro reservado y posición</i>	68
<i>Notificación de préstamos</i>	69
<i>Recomendación de ejemplares nuevos o antiguos, acorde a los gustos y preferencias del usuario</i>	70
PROTOCOLOS DEL AGENTE CAFETERÍA/COMEDOR	72
<i>Notificación de localización de un amigo.</i>	72
<i>Notificación de reserva de mesas.</i>	73
<i>Recordatorio de reserva de mesas.</i>	74
<i>Recomendación de menús.</i>	75
<i>Recomendación de desayunos</i>	76
<i>Recomendación de consumiciones.</i>	78
<i>Tablón de anuncios - cambio de turno (solo para alumnos)</i>	79
<i>Tablón de anuncios - cambio de libros (solo para alumnos)</i>	80
PROTOCOLOS DEL AGENTE CENTRO CULTURAL/DEPORTIVO	82
<i>Servicio de reserva de entradas de eventos culturales/deportivos</i>	82
<i>Servicio de reserva de instalaciones</i>	83
<i>Recordatorio de entradas reservadas</i>	84
<i>Recordatorio de instalaciones reservadas</i>	85
<i>Recordatorio de días sin acudir al gimnasio/piscina (solo para alumnos)</i>	86
<i>Recomendación de eventos culturales / Recomendación de eventos deportivos</i>	87
6. EJECUCIÓN DEL ESCENARIO DE USO CON MUSSUS	89
7. CONCLUSIONES	99
PROBLEMAS ENCONTRADOS.	99
ADAPTACIÓN A OTROS DOMINIOS.	101
TRABAJO FUTURO.	102
CONCLUSIÓN FINAL.....	103
8. BIBLIOGRAFÍA.....	104
9. ÍNDICE DE TABLAS.....	106
10. ÍNDICE DE ILUSTRACIONES.....	108
ANEXO A – DIAGRAMAS DE CLASES	110
ANEXO B – DISTRIBUCIÓN DE LA CARGA DE TRABAJO	113
ANEXO C – ESCENARIO DE USO PARA LA PRESENTACIÓN	114

1. Introducción

Computación ubicua.

Se entiende por Computación Ubicua a la integración de elementos informáticos en el contexto de la vida cotidiana de forma que estos elementos sean omnipresentes pero sin destacar por encima de los habituales y pasando lo más desapercibidos posible. Dicho de otra manera, un sistema ubicuo será aquel que preste servicios a las personas, ayudándoles en la realización de sus tareas habituales pero de manera “amigable”, sin que se vean perturbadas por los dispositivos inteligentes del sistema. Debido a las características anteriormente mencionadas, la Computación Ubicua recibe también el nombre de Inteligencia Ambiental.

El concepto de la Computación Ubicua nace en 1988 y su autoría se atribuye a Mark Weiser[1], informático y director científico de Xerox PARC fallecido en el año 1999, solo once años después de su invención.

Antiguamente los sistemas estaban centralizados y eran compartidos por mucha gente. Ahora nos encontramos en la era de la computación personal, en la cual una persona se comunica con una máquina mediante el escritorio. Según Mark Weiser, con la Computación Ubicua acaba de comenzar la tercera era de la computación, en la que una sola persona controla varias máquinas y la tecnología pasa a un segundo plano de nuestras vidas.

“Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.” Mark Weiser.

En una de sus charlas, Weiser enumeró los cuatro puntos fundamentales que describen a la Computación ubicua:

1. El objetivo de un ordenador es ayudarte a hacer algo.
2. El mejor ordenador es un silencioso e invisible sirviente.
3. Cuanto más puedas hacer mediante la intuición, más inteligente eres; el ordenador debe ampliar tu conocimiento.
4. La tecnología debe crear la calma.

Weiser utilizaba con frecuencia la *Realidad Virtual* para tratar de explicar sus ideas por contraposición y comparación. Decía que las tecnologías ubicuas son opuestas a la Realidad Virtual ya que ésta trata de poner a las personas dentro de un mundo generado por ordenador, mientras que las tecnologías ubicuas fuerzan al ordenador a que conviva con las personas en el mundo real. Para ello, proponía que los ordenadores personales fueran sustituidos por ordenadores invisibles encajados en objetos de uso diario. Cuando en 1991 propuso esta teoría, el nivel de tecnología era demasiado costoso y la Xerox descartó inmediatamente realizarlo. Sólo en 1998 comenzaron a aplicar su teoría, que no pudo ver concluida debido a su muerte un año más tarde.

Para terminar y en relación con el tema de la invisibilidad, cabe destacar un artículo publicado por Mark Weiser en el año 1994 titulado *The world is not a desktop* (El mundo no es un escritorio) en el que explica la importancia de que las tecnologías sean invisibles.

“What is the metaphor for the computer of the future? The intelligent agent? The television (multimedia)? The 3-D graphics world (virtual reality)? The StarTrek ubiquitous voice computer? The GUI desktop, honed and refined? The machine that magically grants our wishes? I think the right answer is “none of the above”, because I think all of these concepts share a basic flaw: they make the computer visible.” The world is not a desktop, Mark Weiser[2].

Agentes.

Según Maes[3] los agentes son sistemas computacionales que habitan en ambientes dinámicos complejos, sienten y actúan autónomamente en ese entorno y al hacerlo llevan a cabo un conjunto de objetivos o tareas para los que fueron diseñados.

Sin embargo para Jennings[4] un agente es un sistema computacional que está situado en un determinado entorno, y que es capaz de forma flexible y autónoma de efectuar acciones sobre ese entorno para alcanzar sus objetivos.

Existen múltiples definiciones de lo que es un agente, dependiendo de a quien se le pregunte. Aunque a grandes rasgos, todas vienen a ser lo mismo.

Un agente es una entidad capaz de resolver problemas que se sitúa en un entorno particular, del cual recibe entradas que le hacen modificar su estado y comportamiento, generando salidas sobre dicho entorno. Además está diseñado para alcanzar unos objetivos específicos y es autónomo, ya que tiene control sobre su estado y sobre su propio comportamiento. Un agente es capaz de exhibir un comportamiento flexible al tratar de resolver un problema y es proactivo, es decir, capaz de modificar sus metas de forma oportunista y tomar la iniciativa para poder alcanzar sus objetivos. Si es necesario, también tiene la posibilidad de comunicarse con otros agentes.

Por tanto, las principales características de un agente son:

- Autonomía y control de sus propias acciones.
- Interacción con el entorno.
- Tendencia a cumplir los objetivos predefinidos por el diseñador.
- Persistencia.
- Capacidad de actuar de forma reactiva y también proactiva.
- Cierta nivel de ‘inteligencia’

Pero también existen otras características muy deseables como la actividad social (comunicación, colaboración y coordinación con otros agentes), el aprendizaje, la racionalidad, la flexibilidad, la benevolencia o la veracidad. El tener una u otra característica depende del tipo de agente, ya que estos pueden ser agentes colaborativos, personales, agentes de interfaz, agentes móviles, agentes informativos, etc.

En cualquier caso, el funcionamiento general de todo agente consiste en que este percibe las entradas que necesita del entorno, las procesa y da como salida las acciones sobre dicho entorno que se deben realizar.

Para llevar a cabo su implementación, los agentes suelen construirse de acuerdo al paradigma de la orientación a objetos, ya que un objeto puede asociarse fácilmente a un agente aunque de manera muy simplificada.

En ocasiones, los agentes no actúan solos. Cuando varios agentes trabajan juntos de manera coordinada dentro de un mismo entorno, nos encontramos con un Sistema Multiagente (SMA).

En otras palabras, un Sistema Multiagente es un conjunto de agentes autónomos, generalmente heterogéneos y potencialmente independientes, que trabajan en común resolviendo un problema. Estos agentes suelen ser capaces de comprometerse con metas comunes, tomar la iniciativa, cooperar y negociar entre ellos e, incluso, compartir conocimiento los unos con los otros.

La rama de la informática dentro de la cual se encuadran estos sistemas recibe el nombre de Inteligencia Artificial Distribuida.

JADE.

JADE^[5] (Java Agent DEvelopment Framework) es un software implementado en lenguaje Java que simplifica la implementación de los sistemas multi-agente a través de un *middleware* que cumple con las especificaciones FIPA y de un conjunto de herramientas gráficas que soportan el despliegue y las fases de depuración. Los agentes de la plataforma pueden estar distribuidos entre varias máquinas que pueden ser fijas o móviles, según sean servidores o usuarios, y ni siquiera necesitan compartir el mismo Sistema Operativo. La configuración se puede controlar a distancia a través de una interfaz gráfica de usuario y puede hacerse incluso en tiempo de ejecución. JADE está completamente implementado en lenguaje Java y la versión mínima requerida para su ejecución es la 1.4.

La sinergia entre la plataforma JADE y las librerías de la plataforma de expansión LEAP permite obtener una plataforma de agentes compatible con el estándar FIPA y con entornos Java móviles inferiores a J2ME-CLDC MIDP 1.0. Las bibliotecas LEAP han sido desarrolladas con la colaboración del Proyecto LEAP y se pueden descargar como un complemento de JADE desde su sitio web oficial: <http://jade.tilab.com/>.

El software JADE es gratuito y lo distribuye Telecom Italia, que es el titular de los derechos de autor, como software de código abierto bajo los términos de la LGPL (Lesser General Public License versión 2). Desde mayo de 2003, se ha creado un consejo que supervisa la gestión del proyecto JADE. Actualmente, dicho consejo consta de cinco miembros: Telecom Italia, Motorola, Whitestein Technologies AG, Profactor GmbH, y France Telecom R&D.

FIPA (Foundation for Intelligent Physical Agents) es una organización sin ánimo de lucro registrada en Ginebra (Suiza) cuyo objetivo es la promoción de las nuevas tecnologías basadas en agentes. Su meta es proporcionar las especificaciones estándar que maximicen la interoperabilidad entre los servicios, los equipos y las aplicaciones basadas en agentes. Esto se consigue mediante la colaboración abierta internacional de las organizaciones miembros, que son empresas y universidades que trabajan en el campo de los agentes. Las especificaciones de FIPA están disponibles públicamente. No se trata de una tecnología para una aplicación específica, sino de unas tecnologías genéricas para diferentes áreas de aplicación. Tampoco es una tecnología independiente, sino un grupo de tecnologías básicas que pueden ser integradas por los desarrolladores para desarrollar sistemas complejos con un alto grado de interoperabilidad.

JADE implementa todas las especificaciones básicas de FIPA que proporcionan el marco dentro del cual los agentes FIPA pueden existir, funcionar y comunicarse.

La última versión de JADE es la JADE 3.6.1 cuya fecha de salida fue el 4 de Noviembre de 2008, pero el desarrollo de JADE continúa. Nuevas mejoras, equipamientos e implementaciones ya han sido previstos, la mayoría de ellos en colaboración con usuarios interesados en la comunidad de JADE.

Metodología GAIA.

GAIA[6] es una metodología que indica las pautas a seguir a la hora de llevar a cabo el desarrollo de un Sistema Multiagente, tanto para la fase de análisis como para la de diseño.

Esta metodología surge de la necesidad de tener alguna técnica de ingeniería de software adaptada específicamente para este tipo de sistemas, ya que las técnicas de ingeniería del software anteriores a ella no son adecuadas para realizar esta tarea. Particularmente, estas técnicas no eran capaces de capturar características como, por ejemplo, la flexibilidad, la autonomía, la capacidad de interacción o la complejidad propias de un agente.

Aun siendo diseñado para trabajar con Sistemas Multiagente, GAIA tiene algunas limitaciones que deben cumplirse para poder ser aplicada, estas son:

- Los agentes son sistemas computacionales de grano grueso, de modo que cada uno hace un uso importante de sus recursos computacionales (como si cada agente tuviese los recursos de un proceso unix).
- Se supone que el objetivo de los agentes debe ser maximizar el beneficio global del sistema aunque esto implique minimizar el propio beneficio del agente. GAIA no está pensado para sistemas en los que existan conflictos reales entre los agentes.
- Los agentes son heterogéneos, de modo que los diferentes agentes podrán ser implementados con diferentes lenguajes de programación, arquitecturas, y técnicas. No se realizan supuestos sobre la plataforma.
- La estructura organizativa del sistema es estática por lo que las relaciones entre los agentes no cambian en tiempo de ejecución.
- Las habilidades y los servicios de los agentes también son estáticos y no cambian en tiempo de ejecución.
- El sistema debe tener un número relativamente pequeño de los distintos tipos de agentes, siendo este menor de cien.

Los principales conceptos manejados por GAIA se dividen en dos categorías: abstractos y concretos.

Los conceptos abstractos son aquellos que se utilizan durante la fase de análisis para conceptualizar el sistema, pero que no necesariamente tienen porque tener alguna correspondencia directa en el sistema. Estos conceptos son: roles, permisos, responsabilidades, protocolos, actividades, propiedades de vitalidad y propiedades de seguridad.

Por el contrario, los conceptos concretos son aquellos que se usan en el proceso de diseño y que suelen tener una correspondencia directa en el sistema real. Estos conceptos son: tipos de agentes, servicios y conocidos.

Un rol define lo que se espera que un agente haga, refiriéndose tanto al objetivo general como en relación a otros agentes. Es decir, que concreta de forma intrínseca la posición del agente en la organización y lo asocia a unos comportamientos establecidos.

Los roles quedan definidos por cuatro atributos ya mencionados: responsabilidades, permisos, actividades y protocolos. Las responsabilidades determinan la funcionalidad y son, con toda probabilidad, el atributo clave de un rol.

Estas responsabilidades se dividen en propiedades de vitalidad y de seguridad. Las propiedades de vitalidad son aquellas que determinan lo que debe lograr un agente dadas unas condiciones de entorno determinadas, mientras que las propiedades de seguridad son aquellas que evitan que algo malo suceda.

En cuanto al resto de atributos, los permisos son los derechos asociados a un rol y determinan los recursos que están disponibles para cada uno, con el fin de que este pueda llevar a cabo sus responsabilidades. Las actividades son los cálculos asociados al rol y que pueden ser llevados a cabo por el agente sin interactuar con otros agentes. Y, finalmente, los protocolos determinan la forma mediante la cual los un rol puede interactuar con el resto de roles.

Una vez definidos los principales conceptos, GAIA proporciona al desarrollador varios modelos tanto para la fase de análisis como para la de diseño.

Para la fase de análisis GAIA propone dos modelos:

- **Modelo de roles:** este modelo identifica los roles claves del sistema. En este modelo, un rol puede ser visto como una descripción abstracta de las funciones que debe realizar una entidad. Para cada rol se definirán sus cuatro atributos ya mencionados: actividades, protocolos, permisos y responsabilidades.
- **Modelo de interacción:** este modelo detalla las dependencias y relaciones entre los distintos roles de la organización multiagente. Para ello, consta de un conjunto de definiciones de protocolo, uno para cada tipo de interacción entre las funciones de los roles. En este modelo, un protocolo puede ser visto como un patrón de interacción que ha sido definido formalmente y fuera de cualquier secuencia de la ejecución, centrando la atención en la naturaleza y la finalidad esencial de la interacción.

Para la fase de diseño los modelos propuestos por GAIA son:

- **Modelo de agentes:** este modelo tiene como objetivo documentar los distintos tipos de agentes que serán utilizados en el sistema en desarrollo, así como las instancias que representaran a dichos agentes en tiempo de ejecución. Para ello, deberá realizarse una asociación entre roles y agentes, que suele ser una relación uno a uno, aunque en muchos casos un agente puede englobar varios roles.
- **Modelo de servicios:** el objetivo de este modelo es identificar los servicios asociados a cada agente y especificar las principales propiedades de estos servicios. Un servicio es a una función del agente, algo así como un método en la terminología de la orientación a objetos pero sin estar disponible para los demás agentes. En otras palabras, es un único y coherente bloque de actividad en la que un agente participará. Las propiedades de un servicio son las entradas, salidas, pre-condiciones y post-condiciones.
- **Modelo de conocidos:** este último modelo es el más sencillo y, únicamente, define los enlaces de comunicación que existen entre los distintos tipos de agente, pero sin profundizar en lo que se envían ni en cuando lo hacen. De este modo, se podrán identificar los posibles cuellos de botella en la comunicación, que pueden causar problemas en tiempo de ejecución. Dichos enlaces se representan mediante un grafo.

2. Estado del Arte

En primer lugar se realizará un análisis sobre el estado del arte en el campo de la Computación Ubicua, así como del uso de agentes para el desarrollo e implementación de los sistemas analizados. Los sistemas que a continuación se comentan se corresponden, en su mayoría, con proyectos de investigación de distintas universidades de todo el mundo.

El primer sistema estudiado es el proyecto **ODISEA**[7] (Universidad Autónoma de Madrid, 2000). Este proyecto tiene la finalidad de desarrollar un entorno inteligente con el cual se pueda interactuar empleando lenguaje natural. Este entorno doméstico será de ámbito ofimático y consistirá en una habitación inteligente, cuyo prototipo ya ha sido creado, que tendrá en todo momento constancia de los ocupantes que hay en ella, de las preferencias de cada uno y de las tareas que están realizando. Haciendo uso de esta información, la habitación podrá realizar acciones que sirvan de ayuda para sus ocupantes de modo que, además, podrá interactuar con ellos mediante diálogos en lenguaje natural.

El sistema hará uso de diálogos específicos para cada una de las posibles necesidades de los usuarios, así como un gestor de diálogos que decida cual de ellos emplear en cada momento, cuando cambiar de uno a otro o que información se intercambia entre ellos.

Las interacciones que podrán llevarse a cabo son el soporte a la colaboración y la comunicación, el acceso a la información y el seguimiento y control del entorno.

La arquitectura básica de ODISEA consta de dos capas: la capa de interacción y la capa de contexto.

La capa de interacción con el entorno físico incluye el conjunto de dispositivos que componen la infraestructura del entorno inteligente. Para llevar a cabo el control de dispositivos, sensores y actuadores se emplea el bus europeo EIB y para el flujo continuo de información se utiliza una red Ethernet que conecta los dispositivos entre sí.

La capa de contexto ofrece información sobre el estado del entorno, sus ocupantes, las actividades que están realizando y las metas que persiguen, etc. Para su implementación se utiliza una estructura de datos centralizada, denominada pizarra y, tanto para la definición de todos los componentes del entorno como para la interfaz con la pizarra, se utiliza el lenguaje XML.

La comunicación con el sistema ODISEA por parte del usuario se lleva a cabo mediante el habla, utilizando expresiones no restringidas en lenguaje natural; es decir expresiones libres y generales, sin restringirse a comandos específicos. A su vez, el entorno responde al usuario utilizando expresiones adecuadas.

Para realizar el reconocimiento de voz, se emplea la herramienta comercial ViaVoice de IBM, haciendo uso también de su capacidad Text to Speech para la síntesis de voz; y la comunicación con el reconocedor se realiza utilizando la Java Speech API. Además, para que el módulo de diálogos pueda procesar la información que se recibe del reconocedor se utilizan un conjunto de analizadores y gramáticas integradas en las herramientas MACO+ y Relax.

En cuanto al problema de hacer que los diálogos sean libres y sin restricciones se tienen en cuenta tres factores fundamentales:

- El usuario se encuentra dentro de un entorno restringido que permite poder “intuir” qué tipo de comunicación establecerá con el sistema.

- Aunque el usuario puede establecer múltiples conversaciones con el sistema dentro de un mismo entorno, estas conversaciones están dirigidas por el sistema y el usuario siempre se encuentra dentro de algunos de los diálogos que tiene previstos el sistema.
- El sistema cuenta con una información de contexto muy rica que se encuentra en la pizarra.

Por tanto, cuando el usuario dice algo, el sistema realiza en primer lugar un análisis morfológico y sintáctico de la expresión que ha pronunciado y, una vez hecho esto, se analiza el resultado para determinar qué ha dicho, apoyándose si es necesario en la información almacenada en la pizarra.

Este sistema consta de un modelo de gestión de diálogos que se basan en la idea de guiones de conversación y plantillas de tarea: los guiones de conversación se componen de una secuencia de patrones de palabras, con sus respectivas relaciones, y de sus respuestas correspondientes. Por otro lado, las plantillas de tarea definen los parámetros requeridos para completar una determinada tarea.

Todos los diálogos que soporta el entorno discurren en paralelo y están controlados de forma centralizada por el supervisor de diálogos. Su funcionamiento sigue los siguientes pasos:

1. El supervisor de diálogos se encarga de recibir el texto del reconocedor y analizarlo para, a continuación, enviar esta información a todos los diálogos que posee el entorno.
2. Cada diálogo compara el texto etiquetado con sus guiones de conversación y responde al supervisor si considera que la expresión pronunciada por el usuario pertenece a su diálogo.
3. El supervisor de diálogos da prioridad al diálogo que se encontraba activo en ese momento o, en su defecto, da paso al siguiente diálogo que ha optado por el turno.
4. El diálogo que recibe el control por parte del supervisor es el que se encarga de ejecutar las acciones descritas en su guión de conversación.

De este modo, se pretende conseguir que resulte sencillo crear nuevos diálogos y añadirlos, o modificar y mejorar los que ya existen. El usuario puede saltar de un diálogo a otro y el sistema sigue la pista de qué acción se pretende realizar en cada momento.

El gestor de diálogos desarrollado dentro del proyecto ODISEA se encuentra en sus primeras fases de su desarrollo. Los módulos aquí expuestos se encuentran implementados, en proceso de construcción o se espera que funcionen en un breve plazo de tiempo. Además, el prototipo está en funcionamiento, y va a permitir en el futuro investigar en diversas áreas de interés relacionadas con las tecnologías del habla. El objetivo es implementar este prototipo en los despachos de varios de los participantes en el proyecto, y empezar a emplearlo como herramienta de trabajo diario.

Otra herramienta de aplicación diaria es el **Modelo interactivo ubicuo aplicado al patrimonio natural y cultural del área del Montsec**[8] (Universidad de Lleida, 2002) que es un proyecto que consiste en la creación de un entorno multimedia móvil y sensible al contexto, aplicado al patrimonio natural y cultural del área del Montsec y esta siendo desarrollado por el Departamento de Informática e Ingeniería Industrial de la Universidad de Lleida desde el año 2002.

Las sierras del Montsec se sitúan en el sector más occidental del Prepirineo catalán (Lleida). Este paraje cuenta con una extraordinaria riqueza geológica, histórica, paleontológica y paisajística, por lo que dispone de un patrimonio natural y cultural de gran importancia que lo convierten en un laboratorio natural muy apreciado.

Mediante la aplicación de este modelo al entorno del Montsec se ayudará a preservar y difundir los bienes y recursos naturales y culturales del parque, obteniendo en el área un verdadero "espacio interactivo". Para ello cualquier visitante que disponga de un terminal podrá ser identificado como usuario y se le ofrecerá todo tipo de información adaptada a sus necesidades de consulta y guía interactiva personalizada que le puedan ir surgiendo a lo largo de su itinerario, según el perfil, posicionamiento geográfico e historial del usuario.

Concretamente, los parámetros que serán tenidos en cuenta en referencia al usuario son su posicionamiento geográfico, sus características personales, su historial de visitas y las actividades que ha realizado anteriormente.

La información a presentar al usuario estará, por tanto, adaptada a estos aspectos que la aplicación debe obtener de forma automática.

Así pues, además de las interacciones entre usuario, aplicación y agente de la interfaz, se establecen otras dos interacciones: una entre la aplicación y el entorno y otra entre la aplicación y el contexto.

La interacción entre la aplicación y el entorno consiste en que, en función del posicionamiento geográfico del usuario, el sistema realiza un proceso de búsqueda de la información correspondiente al sector involucrado. Por otro lado, en la interacción entre la aplicación y el contexto intervienen el historial de las visitas y actividades realizadas anteriormente por el usuario, así como su perfil (personalidad, preferencias, intereses y características básicas como edad y nivel de estudios, etc.). Haciendo uso de estos aspectos relativos al usuario, se adapta la información seleccionada hasta el momento, para generar y presentar información hipermedia de forma personalizada.

Según lo anterior, otro de los objetivos a conseguir en este modelo es que sea adaptativo, tanto a diferentes usuarios como a diferentes circunstancias y para ello se han establecido diferentes perfiles patrón con los que poder catalogar a los posibles visitantes de la zona, dentro de un conjunto de categorías.

Además, la zona interactiva no está limitada a una pantalla como en otros sistemas, ya que la aplicación, consciente de su entorno, reacciona y responde ante la detección de nuevas coordenadas geográficas, mostrando la información que corresponde al área visitada.

La arquitectura del sistema está dividida en tres capas: capa de datos o ítems de entrada, capa de núcleo de la arquitectura y capa relativa a la interfaz.

El sistema cuenta con una Base de Datos Multimedia (BD-MM) donde se almacena el contenido de todos los bloques temáticos, conteniendo información multidisciplinar sobre el ámbito de la geografía, historia, geología y biología del Montsec.

Además, se dispone de un Sistema Basado en el Conocimiento (SBC), como módulo destinado para resolver el razonamiento sobre el problema, de modo que el perfil de usuario y su historial serán modelados dinámicamente como hechos de este SBC.

Por otro lado, se cuenta con un módulo de Navegación Dinámica (ND) que se encarga de elaborar la interfaz teniendo en cuenta la información de la BD-MM, ya que tanto dicha interfaz como la navegación a través de ella deben ser también personalizadas al usuario.

En cuanto a la presentación de datos, esta se realiza empleando XML ya que, al tratarse de un formato universal, cualquier tipo de terminal del que disponga el usuario podrá emplear el sistema (PC, PDA, TV, WAP o un teléfono móvil de tercera generación con tecnología UMTS). Para convertir la información seleccionada en este formato XML se emplea un servidor JSP.

Finalmente, a la hora de desarrollar el sistema se han empleado dos técnicas: prototipado y escenarios de futuro.

Y, continuando con los prototipos, unos años después surge una nueva herramienta denominada **MobileEye**^[9] (Universidad de Deusto, 2004). MobileEye es la primera aplicación prototipo, desarrollada por la Universidad de Deusto para evaluar el concepto de EMI2.

En primer lugar es necesario explicar este concepto. EMI2 es una arquitectura que permite avanzar hacia el concepto de inteligencia ambiental o computación sensible, utilizando el terminal móvil como catalizador y sus principales características son que almacena todas las interacciones del usuario para extraer los patrones de comportamiento, en un almacén que puede ser local o remoto. Además utiliza mecanismos de localización (GPS, Bluetooth, reconocimiento de patrones (código barras)) y mecanismos de descubrimiento del entorno (Bluetooth, infrarrojos, cámara). Las tecnologías en que se basa son los servicios web y XML.

Los elementos básicos de EMI2 son el EMIBehaviourRepository (almacén de datos de interacción), el EMIDevice (dispositivo del entorno) y el EMIProxy (representante del usuario propietario del terminal móvil) que sondea la presencia de EMIDevices.

Haciendo uso de una simplificación de esta arquitectura, MobileEye emplea una combinación de códigos de barras 2-D y móviles con cámara, para que éstos reconozcan los objetos de su entorno, añadiendo “capacidad de visión” al teléfono. Además, estos objetos del entorno son aumentados con representantes software (proxies) en forma de servicios web.

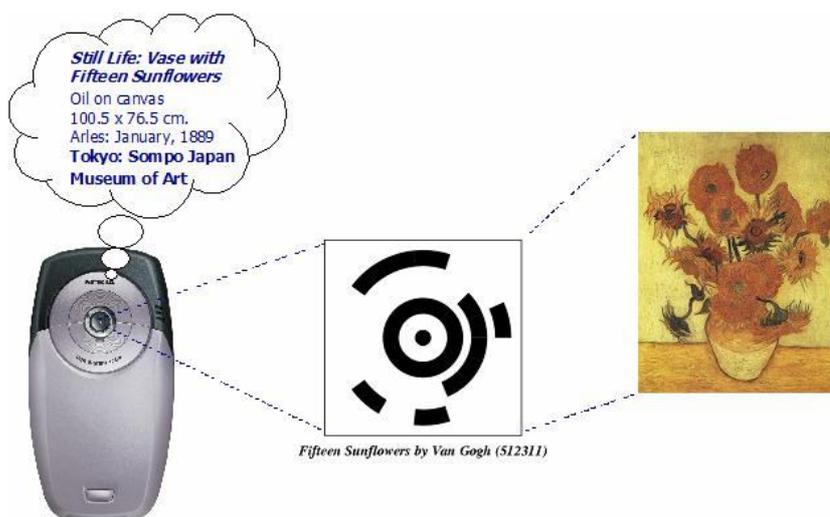


Ilustración 1 - Funcionamiento de MobileEye

Cuando el terminal reconoce un objeto en el entorno, presenta información sobre este y una interfaz a través de la cual el usuario puede operar con el servicio web del objeto etiquetado.

Los códigos de barras empleados en MobileEye para reconocimiento pueden ser de tres tipos: el código TRIP consistente en un código de barras 2-D con código ternario, un código de 2 anillos de codificación de 16 bit o el Ojo de Buey, que es muy sencillo de identificar puesto que no varía con respecto a la rotación o la perspectiva y además posee un contraste alto.

MobileEye dispone de dos modos de funcionamiento, el primero de ellos es a través de un programa Java para dispositivos embebidos basado en J2ME (MIDlet). En este modo, la aplicación captura imágenes y las envía a petición del usuario al servidor de MobileEye. A continuación, dicho servidor procesa la imagen y envía como respuesta un fichero XML con información sobre el objeto reconocido y una url apuntando al servicio web que lo representa. Finalmente, el MIDlet MobileEye visualiza la respuesta y/o la interfaz del servicio web del objeto etiquetado. Para todo ello se requiere un teléfono móvil que soporte J2ME MMAPI.

En el segundo modo de funcionamiento el usuario simplemente utiliza capacidades ya disponibles en móvil. Para ello, deberá enviar un MMS con la imagen del objeto etiquetado a la dirección mobileeye@ctme.deusto.es. A continuación el servidor de MobileEye comprueba la recepción de esos mensajes en el servidor mail y extrae de ellos la imagen para después procesarla y enviar la respuesta. Este modo está disponible para un gran abanico de teléfonos móviles de última generación.

Ese mismo año surge una propuesta de tesis con título **Sistema de control cooperativo multi-agente en el entorno de una vivienda asistida**^[10] (Universidad Politécnica de Cataluña, 2004). Este sistema, desarrollado por la Universidad Politécnica de Cataluña, consiste en la integración de los distintos elementos de una vivienda, para poder asistir y controlar a distancia el estado de personas mayores o pacientes que viven solos.

Este proyecto tiene dos finalidades fundamentales. En primer lugar, construir una arquitectura multi-agente sobre una red de Elementos Hardware Inteligentes (EHI), de modo que los distintos elementos de la casa (sensores, actuadores, sensores-actuadores o grupos de estos) actuarán como agentes independientes. Para ello, cada dispositivo, o conjuntos de estos, dispondrá de un procesador que, haciendo uso de la información y datos que posee, será capaz de tomar decisiones en cuanto a que acciones llevar a cabo. Además se dispone de un bus de comunicaciones mediante el cual los dispositivos pueden intercambiar información que pueda ser relevante para otros.

El procesado de la información para la toma de decisiones se hará haciendo uso de reglas y/o marcos o mediante sistemas neuro-borrosos o de soft-computing, según sea el nivel crítico de la tarea a realizar.

La segunda finalidad del proyecto es realizar la integración de este sistema en una vivienda asistida de modo que la interacción con el usuario sea lo más amigable posible. Para asegurarse esto, se emplearán mecanismos de comunicación inalámbrica y vía web, así como las correspondientes interfaces de usuario. La opinión del personal médico y clínico será muy valorada a la hora de decidir que mecanismos son adecuados para cada paciente concreto.

A la hora de procesar la información, este comportamiento debe ser adaptativo, de modo que el propio sistema deberá aprender a mejorar el rendimiento de la vivienda asistida. Para ello existen distintos modos, como puede ser introduciendo conocimiento nuevo (hechos, comportamiento, reglas) en el sistema, generalizando conceptos a partir de múltiples ejemplos, especializando conceptos a partir de ejemplos concretos, reorganizando la información dentro del sistema de forma más eficiente, creando o descubriendo nuevos conceptos y reutilizando la experiencia.

En cuanto a las comunicaciones, el paciente debe verse perturbado lo menos posible, de modo que es necesario minimizar el número de cables de conexión entre el PC y los elementos hardware inteligentes. Por ello, se utilizará un bus CAN para comunicar los dispositivos entre si y para la comunicación entre las distintas estancias de la vivienda se utilizará una red PLC que, al utilizar la línea eléctrica, no necesita instalación de nuevos cables. Otra alternativa es la tecnología Bluetooth que además de suponer un coste menor, resulta de mucha utilidad para los componentes móviles, que no pueden depender de cables.

Para las interfaces fijas se emplearán dispositivos de pantalla táctil y, para las portátiles, PDAs y teléfonos móviles con tecnología Bluetooth.

Finalmente, para seleccionar los dispositivos adecuados para iniciar el proyecto se ha contado con la colaboración de expertos del Hospital Sant Antoni Abat y, conjuntamente, se tomó la decisión de que los dispositivos indispensables serían un holter o registrador de señal cardiaca, matrices de sensores de presión y acelerómetros, además de otros elementos para la monitorización y control de pacientes.

Y en este mismo campo, encontramos otra propuesta llamada **Computación ubicua para el cuidado de pacientes en estado crítico en el hogar**^[11] (Universidad Autónoma de Baja California, 2004). Este proyecto hace relación con una propuesta de tesis realizada por Pedro Santana en el año 2004, para la Facultad de Ingeniería de la Universidad Autónoma de Baja California (UABC) y tiene muchas similitudes con el sistema anterior.

Este proyecto guarda relación con la telemedicina en el hogar (THC por sus siglas en inglés –Telehomecare). Básicamente consiste en un sistema que garantiza que si le ocurre algo al paciente cuando se encuentra en su hogar se pondrá en marcha una red de recursos para ayudarlo lo antes posible. Esta idea es muy valorada por los usuarios pero, sobre todo, por sus familiares.

En primer lugar, es necesario saber que existen tres tipos de THC. El primero de ellos es el modelo básico, o modelo de tele-alarma, que utiliza teléfonos de emergencia y sistemas de alarma personales que, con solo presionar un botón, contactan de forma automática al usuario con el centro de asistencia. El segundo modelo parte del anterior pero utiliza sistemas inteligentes que toman decisiones automáticamente y son capaces de hacer sonar la alarma ante la sospecha de que algo malo puede estar ocurriendo, sin necesitar que el usuario intervenga. El último modelo consta de comunidades virtuales, donde se reúnen usuarios y profesionales para ofrecer los últimos consejos sobre prevención o guías para tratar desde casa algunos síntomas de enfermedades comunes.

Dos características fundamentales para el sistema son la Ubicuidad y la Transparencia y este debe ser consciente del contexto en el que se encuentra.



Ilustración 2 - Ejemplo de habitación en THC

Según lo dicho anteriormente, el proceso de investigación del proyecto parte de tres hipótesis. La primera de ellas es que el segundo modelo de THC es mejorado mediante tecnología de cómputo ubicuo y conciencia del contexto. Además el realizar un caso de estudio para generar escenarios de uso permite crear aplicaciones de THC útiles. Y finalmente, los agentes autónomos permiten abordar todas las complejidades de un sistema ubicuo THC.

Por tanto, el objetivo general de este proyecto de investigación es el de diseñar ambientes de cómputo ubicuo en el sector salud, identificando un escenario de uso real, así como el desarrollo y evaluación de un prototipo.

Finalmente, de este estudio pueden extraerse una serie de limitaciones que también pueden aplicarse a proyectos similares, estas son:

- Gran parte del éxito de los sistemas THC, así como de otros sistemas similares, depende de que los pacientes o usuarios acepten usarlos, una limitación sería que estos no acepten el uso de la tecnología propuesta.
- El éxito de los sistemas THC no sólo depende de la aceptación de uso de los pacientes, si no también de los médicos y expertos en salud, por lo que también sería una limitación el hecho de que no se acepte su uso por este sector importante.
- El no poder abordar todas las complejidades de un sistema ubicuo de THC con agentes autónomos limitaría en el sentido de que habría que buscar otras alternativas.
- En el cómputo ubicuo la tecnología se encuentra en todas partes, por lo que es necesario contar con el equipo suficiente para crear ambientes ubicuos; por tanto una limitación podría ser no obtener los recursos necesarios para conseguir este tipo de equipo.

Cambiando de escenario, nos encontramos con el sistema **AMICO**[12] (Fundación TEKNIKER, 2004). AMICO es un proyecto real desarrollado en el Laboratorio de Inteligencia Ambiental de TEKNIKER en el año 2004 que consiste en la creación de una arquitectura multiagente que es capaz de seguir al usuario en su recorrido por el Laboratorio para ofrecerle la información que necesite en cada momento y en el dispositivo más adecuado para cada caso.

En primer lugar cabe destacar que uno de los requisitos para que la Inteligencia Artificial (Aml) ofrezca todas sus potencialidades radica en la necesidad de que el “ambiente” reconozca la presencia de las personas y pueda ubicarlas en un contexto, tanto geográfico como de actividad. Esto permite ofrecer la información necesaria para el desarrollo de una actividad.

También recordar que las aplicaciones de Aml exigen la integración en el “contexto” cotidiano de aplicaciones computacionales de forma amigable y no intrusiva. En la definición de este contexto cabe referirse a cualquier hecho relevante en el entorno, como la posición del usuario y parámetros asociados a la localización, los dispositivos al alcance de la persona (teléfono, PDA, etc), las preferencias del usuario y las personas y servicios en el entorno del sujeto

Según lo mencionado, el departamento de Ingeniería de Producción de TEKNIKER ha creado un Sistema Multiagente habilitado en uno de sus laboratorios, cuya finalidad es crear un espacio de inteligencia ambiental en el ámbito de la fabricación, que sea capaz de utilizar la información de contexto para anticipar las necesidades de los usuarios y ofrecerles la solución adecuada.

Este laboratorio Aml ofrece un entorno experimental cuyo elemento central es una fresadora de alta velocidad, con motores lineales y cabezal de sustentación magnética complementándose con diferentes dispositivos y aplicaciones como: tecnología de interfaces (reconocimiento de voz, “tablet PCs”, PDAs, teléfonos, etc), sistemas de localización de personas y dispositivos, aplicaciones típicas de un entorno de fabricación rediseñadas e implementadas en forma de Agentes, sistema de visión artificial y acceso a redes de información inalámbricas basadas en 802.11 y Bluetooth.

A continuación se puede ver un técnico de mantenimiento de AMICO que emplea HMD y sistema de reconocimiento de voz.



Ilustración 3 - Técnico de AMICO

Una de las funciones principales de AMICO es mostrar información contextual atendiendo a tres criterios: el perfil de usuario, el dispositivo más adecuado, y la ubicación del usuario. Además permite el acceso a funcionalidades de la máquina según la ubicación del usuario y aprende y se adapta a las preferencias de este.

Como Sistema Multiagente, AMICO cumple los siguientes requisitos: movilidad, multiplataforma, autonomía, proactividad, capacidad de comunicación, escalabilidad y capacidad de adaptación al usuario.

Además el laboratorio dispone de un sistema de red que soporta tanto WIFI como Bluetooth y la arquitectura se basa en una plataforma JADE (Java Agent DEvelopment Framework) en la que conviven diversos agentes distribuidos en los diferentes dispositivos y para cuya comunicación emplean mensajes FIPA-ACL.

Los agentes que intervienen en el sistema son: el Agente Broker, el Agente Traker, el Agente Móvil y el Register.

El Agente Broker controla y gestiona la información de contexto del Laboratorio encargándose de ofrecer al usuario la información adecuada a ese contexto. El Agente Traker notifica a otros agentes la localización de personas, sistemas con capacidad de computación o cualquier elemento localizable. El Agente Móvil se encarga de mostrar información “contextualizada” en los diferentes dispositivos disponibles, siendo capaz de moverse de uno a otro. Y el Register registra los diversos dispositivos existentes en la plataforma de agentes del Laboratorio.

Como conclusión, este Sistema Multiagente ha dado muy buenos resultados, aunque con algunas limitaciones:

- La falta de una metodología a la hora de desarrollar los agentes.
- La necesidad de herramientas de “alto nivel” para la definición de los agentes y su desarrollo.
- La definición de ontologías reutilizables en el ámbito de la Inteligencia Ambiental.
- Mecanismos para la integración de “legacy systems”.

Otro sistema real ya desarrollado es **PUMAS**^[13] (LSR-IMAG Laboratory, 2005). PUMAS (Peer Ubiquitous Multi-Agent System), que es un proyecto que fue publicado en la Revista Colombiana de Computación en el año 2005, consiste en un framework basado en agentes cuyo principal objetivo es proveer a usuarios nómadas con información relevante adaptada a diferentes criterios cuando acceden a Sistemas de Información Web (SIW) a través de sus Dispositivos Móviles (DM). Estos criterios pueden ser sus preferencias, su localización, etc. En cuanto a los SIW, estos pueden ejecutarse en uno o más servidores o, en uno o más DM.

Los agentes de PUMAS están organizados para llevar a cabo la búsqueda de la información que el usuario requiere, proveniente de uno o más SIW, y que debe ser inteligente y adaptativa. Inteligente porque debe estar basada tanto en el conocimiento propio, adquirido e inferido de los agentes, como en su capacidad de razonamiento. Y adaptativa porque toma en cuenta las características del usuario y las de su DM.

Cuando un usuario busca información, a través de su DM, sus consultas se propagan a través de los agentes de PUMAS hacia el Router Agent. Estos agentes de PUMAS, añaden a las consultas algunas características del usuario y de su DM para ser tenidas en cuenta al adaptar la información. Por otro lado, el Router Agent ejecuta el proceso de enrutamiento de consultas, que consiste en la búsqueda de fuentes de información adecuadas que pueden responder a las consultas del usuario teniendo en cuenta sus preferencias, las características de su DM, su localización, etc.

La información acerca del usuario y su DM es almacenada en archivos XML específicos y el conocimiento manejado por los agentes se almacena en Bases de Conocimiento (BC).

La arquitectura de este sistema se compone de cuatro Sistemas Multi-Agente: el SMA de Conexión, el SMA de Comunicación, el SMA de Información y el SMA de Adaptación.

El SMA de Conexión es la plataforma central de PUMAS y provee los mecanismos que facilitan la conexión de diferentes tipos de DM al sistema.

El SMA de Comunicación garantiza una comunicación transparente entre los DM y el sistema y aplica el Filtro de Despliegue que consiste en mostrar al usuario, a través de su DM, la información de acuerdo a las restricciones técnicas de su DM. Para la aplicación de este filtro, cuenta con la ayuda de los agentes del SMA de Adaptación.

El SMA de Información recibe las consultas del usuario y las redirecciona hacia el Sistema de Información (SI) más “adecuado”, aplica el Filtro de Contenido de acuerdo al perfil de usuario en el sistema y retorna los resultados “filtrados” al SMA de Comunicación. Para la aplicación del filtro, cuenta con la ayuda de los agentes del SMA de Adaptación.

Los agentes del SMA de Adaptación se comunican con los agentes de los otros tres SMA con el fin de intercambiar información acerca del usuario, de la conexión y de la comunicación, de las características del DM, etc. Los servicios y tareas que desempeñan consisten, esencialmente, en el manejo de los archivos XML. Los agentes del SMA de Adaptación manejan conocimiento que permite filtrar la información a los usuarios. Existe también conocimiento que es inferido a partir del análisis de la historia del usuario en el sistema, incluyendo sus últimas conexiones, consultas, preferencias, etc.

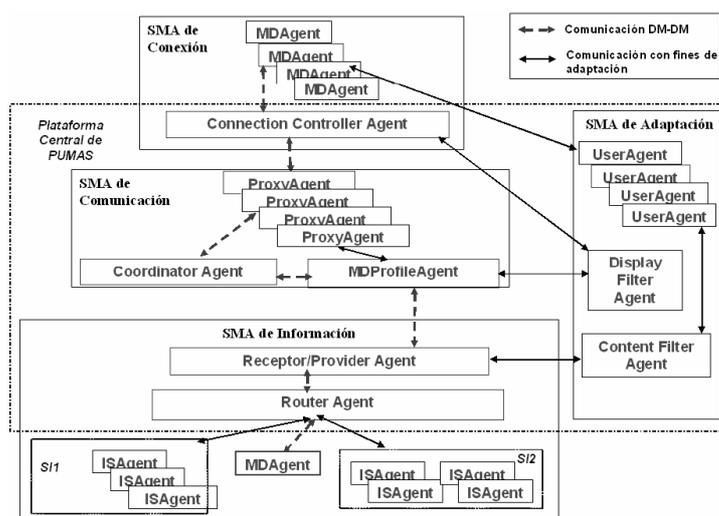


Ilustración 4 - Organización de PUMAS

La implementación de PUMAS se ha llevado a cabo empleando JADE-LEAP, que es una plataforma acorde a los estándares FIPA. Para ello es necesaria la implementación y prueba de cada uno de los SMA por separado.

Para la representación del conocimiento JADE-LEAP utiliza JESS y un conjunto de clases (JADE Content Package) que permiten la definición de ontologías.

Finalmente, el sistema ha sido probado en Pocket PCs con Windows CE, que utiliza Crème, (acorde con la implementación Personal Java) y dos PDA con PalmOS utilizando una implementación MIDP 1.0.

El último sistema estudiado es el más reciente y el que más relación guarda con nuestro propio sistema, ya que también se desarrolla en un contexto universitario, y su nombre es **IUMELA**^[14] (Universidad de Dublín, 2007).

Intelligent Ubiquitous Modular Education Learning Assistant (IUMELA) es un asistente para el aprendizaje diseñado por la Universidad de Dublín, cuya funcionalidad consiste en proporcionar ayuda a los estudiantes durante el proceso de toma de decisiones en el ámbito educacional, mediante el uso de sistemas multi-agente (MAS). Más concretamente, se trata de un asistente inteligente capaz de dar soporte a los estudiantes de tercer grado, a la hora de la elección de módulos de asignaturas basándose en sus preferencias, tanto de aprendizaje como personales, y en sus habilidades académicas.

IUMELA está diseñado de modo que, para su utilización, los alumnos deberán disponer de un dispositivo móvil XDA Mini S con la tecnología integrada. Al tratarse de un dispositivo ligero, este posee limitaciones propias de los dispositivos inalámbricos, como el tamaño y resolución de pantalla, la calidad del color o el tiempo de vida de la batería, por lo que existirán limitaciones en la salida de datos, no pudiendo mostrarse gran cantidad de ellos, o de imágenes, gráficos, etc. Pero por otro lado, al tratarse de dispositivos ligeros e inalámbricos se consigue la principal característica de este sistema, la ubicuidad.



Ilustración 5 - Dispositivo con IUMELA instalado

El sistema consta de cinco tipos de agentes: ayudante (agente que se ejecuta en el cliente del estudiante), moderador (actúa como intermediario), agente de aprendizaje, agente experto y agente de análisis. Y para el intercambio de mensajes entre ellos se emplea ABITS FIPA Messenger, que es un API de Java que permite manejar comunicaciones de un solo sentido desde entornos Java a MAS que implementan FIPA.

En este sistema, cada alumno es identificado unívocamente por un identificador de usuario y una contraseña, de modo que al iniciar sesión, el contexto actual de dicho alumno es actualizado. De este modo, las habilidades académicas y las capacidades personales del alumno son almacenadas junto con el resto de su perfil (identificador de estudiante, nombre, email, etc.). A partir de estos datos almacenados se realizan los pasos necesarios para realizar la recomendación más adecuada para el alumno.

Comparativa y conclusiones

Para finalizar el estado del arte se incluye la siguiente tabla, en la que se resumen las principales características de los sistemas anteriormente comentados, para después poder realizar fácilmente una comparativa entre ellos y nuestro propio sistema a desarrollar. De este modo, podrán extraerse, de manera sencilla, conclusiones que nos ayudarán a la hora de elegir distintas posibilidades de implementación, lenguajes de programación, etc.

Resumen de sistemas:

SISTEMA	DESCRIPCIÓN	TIPO	LENGUAJES Y HERRAMIENTAS	DISPOSITIVOS
ODISEA	Habitación inteligente en el contexto ofimático, que emplea lenguaje natural.	Prototipo.	ViaVoice de IBM Java Speech API XML MACO+ - Relax	Dispositivos inteligentes Bus europeo EIB Red Ethernet
Proyecto Montsec	Entorno multimedia móvil y sensible al contexto, aplicado al patrimonio natural y cultural del área del Montsec.	Prototipo.	BD-MM y SBC XML Servidor JSP	Terminal móvil (PC, PDA, TV, WAP o un teléfono móvil de 3ª gen con Tecn. UMTS)
MobileEye	Aplicación de reconocimiento de imágenes mediante un teléfono móvil.	Prototipo.	XML J2ME MMAPAPI	Teléfonos móviles con cámara (opc. que soporten J2ME MMAPAPI) Infrarrojos Bluetooth GPS Códigos de barras
Vivienda asistida	Vivienda asistida, con elementos inteligentes integrados, para el cuidado de ancianos y pacientes que viven solos.	Propuesta.		Dispositivos inteligentes (Bus CAN y Red PLC) Telfs. móviles con Bluetooth Disp. con pantalla táctil Disp. médicos
Sistema THC	Sistema de Telehomecare, para el cuidado de pacientes en estado crítico en el hogar.	Propuesta.		Dispositivos inteligentes

AMICO	Arquitectura multiagente para ofrecer información en un Laboratorio de Inteligencia Ambiental en Fabricación.	Real.	JADE (FIPA-ACL)	Tecn. de interfaces (rec. de voz, tablet PCs, PDAs, teléfonos, etc.) Sistemas de localización Sistemas de visión artificial Redes 802.11 y Bluetooth
PUMAS	Sistema para proveer a usuarios nómadas con información relevante adaptada a diferentes criterios cuando acceden a SIW a través de sus Dispositivos Móviles.	Real.	BBCC XML JADE-LEAP (JESS)	Terminal móvil (Pocket PCs con Windows CE, PDAs con PalmOS, etc.).
IUMELA	Asistente de aprendizaje para proporcionar ayuda a los estudiantes durante el proceso de toma de decisiones, en el ámbito educacional.	Prototipo.	JAVA (ABITS FIPA Messenger)	Dispositivo móvil XDA Mini S
MUSSUS	Sistema de suministro de servicios en un entorno universitario.	Simulación.	JADE (estándar FIPA) XML	Terminal móvil con Bluetooth o wifi (solo para implantación real)

Tabla 1 - Comparativa de sistemas

Entre los sistemas estudiados podemos encontrar muchas diferencias, no solo en el contexto en el que se desarrollan y sus funcionalidades, sino también en la forma de implementación y en los dispositivos utilizados. De este modo, en cuanto al entorno, tenemos sistemas que van desde los ejemplos clásicos de viviendas asistidas o laboratorios inteligentes, hasta sistemas para proveer información en un entorno natural. Igualmente, en cuanto a la implantación, tenemos sistemas que son meramente propuestas, pero también sistemas implantados y probados realmente con resultados excelentes.

Haciendo referencia a la implementación, a pesar de encontrar algunas diferencias como por ejemplo los tipos de dispositivos, sí encontramos un denominador común que es el uso de JAVA como lenguaje de implementación. Aunque dentro del uso de este lenguaje también existen diferencias ya que, por ejemplo ODISEA emplea una API de Java (Java Speech API) para el reconocimiento de voz, IUMELA emplea otro API que es ABITS FIPA Messenger para manejar comunicaciones de un solo sentido en JAVA, MobileEye emplea J2ME MMAPAPI y sistemas como PUMAS o AMICO emplean JADE.

Otras características comunes a algunos de ellos son el uso de FIPA como estándar para el formato de los mensajes y la utilización de XML para la presentación y almacenamiento de datos.

Haciendo referencia ahora a los dispositivos, podremos encontrar tipos muy variados, ya que el contexto y las funcionalidades de estos sistemas son muy diferentes. Así, podremos encontrar dispositivos médicos (holter, registrador de señal cardiaca, etc.) en el caso de la vivienda asistida, códigos de barras para identificación en MobileEye o sistemas de visión artificial en AMICO, etc.

Pero también hay algo común a casi todos los sistemas, que es el uso de dispositivos móviles por parte de los usuarios del sistema. El uso de Java y XML permite la instalación de las aplicaciones-usuario de casi todos los sistemas en cualquier tipo de dispositivo móvil (teléfono, ordenador, PDA, etc.) aunque algunos de estos especifican que sea en teléfonos móviles, ya que son más ligeros y es más probable que el usuario disponga de uno. Y más concretamente, IUMELA está diseñado para funcionar con el dispositivo móvil XDA Mini S, lo cual puede provocar problemas de implantación, ya que no todos los usuarios podrán disponer de este dispositivo concreto o podrán sentirse reacios a tener que adquirir uno.

Por tanto, basándonos en los sistemas anteriores, MUSSUS será un sistema simulado de suministro de servicios para alumnos y profesores en un entorno universitario, que estará implementado en Java, como casi todos los mencionados, y más concretamente en JADE. Además, sus mensajes seguirán el estándar FIPA, como es usual en mucho de estos sistemas. Finalmente, en caso de ser necesario almacenamiento de datos, se ha optado por la opción XML al igual que en todos los sistemas estudiados. Esta elección de lenguajes permitiría, en caso de implantación real, que el sistema pueda ser instalado en cualquier tipo de dispositivo móvil que disponga de comunicaciones inalámbricas, como Bluetooth.

3. Objetivos y planteamiento del Proyecto MUSSUS

Multi-agent Ubiquitous System to Supply University Services.

¿Qué es MUSSUS?

MUSSUS es un Sistema Multi-Agente que, basándose en el concepto de Computación Ubicua, tendrá como funcionalidad básica el proveer a sus usuarios de los servicios que precisen, dentro de un entorno universitario.

Dado que este proyecto se realiza para la Universidad Carlos III de Madrid, será esta universidad la que sirva como referencia a la hora tanto de seleccionar las zonas en que se dividirá el sistema, como para definir las funcionalidades prestadas por cada una. Estas funciones irán desde avisos de publicación de notas o de vencimientos de plazos entrega, hasta recomendación de menús de comida.

Gracias a este sistema, el usuario tendrá una relación más dinámica con su entorno universitario y a su vez esta relación será más cómoda, ya que el usuario no tendrá que estar pendiente de buscar y obtener los servicios que necesita puesto que, en cuanto el agente servidor de la zona en que se encuentra le haya detectado, le proveerá de todos los servicios que considere oportunos para él.

Para la implementación del sistema se ha considerado que hacerlo como un Sistema Multi-Agente es la mejor opción, ya que el uso de agentes permite conseguir fácilmente la característica más deseada del sistema, la ubicuidad. Esto es así porque los agentes pueden ser fácilmente omnipresentes, ya que pueden distribuirse cómodamente por la universidad tanto en el caso de los agentes servidores, que además verán su trabajo más dividido según la zona y función de cada uno; como en el caso de los agentes usuarios, que además deben ser móviles. La característica de movilidad se consigue cómodamente mediante el uso de agentes instalados en cualquier dispositivo portátil (teléfono, PDA, etc.). De este modo, se consigue también la ventaja de que el sistema no perturbe en exceso a sus usuarios y, sobre todo, a las personas que no son usuarios del sistema y que también se encuentran en el entorno universitario. Además, al implementarse cada servidor de zona como un agente distinto, el sistema puede ampliarse en un futuro fácilmente con nuevas zonas, sin que el resto del sistema se vea afectado lo más mínimo.

Finalmente, cabe explicar que el sistema MUSSUS va a ejecutarse mediante una simulación, empleando una interfaz implementada en lenguaje JAVA. El motivo de esto es que implementar y desarrollar un sistema real en el entorno de la universidad sería muy costoso, para tratarse de un proyecto de final de carrera. Aunque estos costes guardan más relación con la colaboración que se necesitaría por parte de la universidad, los usuarios que tendrían que probar la aplicación y, sobre todo, con los costes temporales. Sin embargo, los costes económicos no serían muy elevados, puesto que cada agente usuario deberá poseer su propio dispositivo móvil y para los agentes servidores bastaría con una maquina para cada uno, con algún tipo de tecnología de comunicación inalámbrica.

Agentes de la aplicación

En términos generales, el sistema dispondrá de dos tipos de agentes, usuarios y servidores, que a su vez se dividirán en distintos tipos según su rol en el caso de los usuarios y según su localización y funcionalidades en el caso de los servidores.

Agentes usuarios

Los agentes usuarios del sistema podrán ser alumnos o profesores.

El sistema proveerá a los usuarios de diferentes servicios, existiendo algunos comunes, pero también otros específicos para cada usuario, según sea su tipo. De modo que, por ejemplo, un alumno podrá recibir notificaciones de publicación de notas que resultarían inútiles y sin sentido para un usuario profesor. Del mismo modo, un profesor podrá recibir una notificación de correo en su casillero, siendo esto imposible para un alumno. Sin embargo ambos podrán, por ejemplo, recibir recomendaciones de menú en el comedor.

Además los usuarios, independientemente de si son alumnos o profesores, dispondrán de:

- NIA o PDI según sean alumnos o profesores respectivamente, para su identificación unívoca.
- Un perfil, que reflejará los datos generales de usuario. Por ejemplo su nombre, fecha de nacimiento, curso, etc.
- Unas preferencias, que reflejarán los gustos del usuario. Por ejemplo que le guste el teatro, o el fútbol, o la literatura, o incluso el queso en la comida. Y también datos como que este buscando a un profesor concreto o que desee un ejemplar de la biblioteca, etc.
- Un historial almacenado por las zonas, que registrará algunas de las actividades realizadas por el usuario. Por ejemplo la asistencia al gimnasio y piscina o los últimos libros extraídos de la biblioteca.

Estos datos serán imprescindibles para el sistema a la hora de tomar decisiones, realizar recomendaciones, etc.

Agentes servidores

Los agentes servidores del sistema serán:

- Secretaría.
- Biblioteca.
- Cafetería.
- Centro Cultural/Deportivo.

Además existe un agente servidor-central llamado Universidad para funciones organizativas y de registro. Básicamente localiza a los usuarios y se encarga de ponerlos en contacto con las zonas en las que se encuentran.

Cuando un agente usuario entra dentro de la zona de un determinado agente servidor, éste detectará la llegada del usuario gracias al agente Universidad y le proporcionará los servicios que considere adecuados acorde a su perfil, preferencias, historial, actualizaciones del sistema, contexto actual, etc.

A continuación se muestran algunas de las funcionalidades que cada uno de los agentes servidores podrá proporcionar.

- Agente Secretaría:
 - Notificación de entregas en el casillero (sólo para profesores).
 - Notificación de reclamaciones (sólo para profesores).
 - Notificación de publicación de notas / Posibilidad de escribir una reclamación a una nota (sólo para alumnos).
 - Recordatorio de plazos de entrega de trabajos/prácticas (sólo para alumnos).
 - Recordatorio de fechas y lugares de exámenes (sólo para alumnos).
 - Notificación de localización de un profesor al que se está buscando.
 - Notificación de localización de un alumno al que se está buscando.
 - Notificación de que un objeto perdido ha sido encontrado.
- Agente Biblioteca:
 - Recordatorio de plazos de entrega de ejemplares (sólo para alumnos).
 - Recordatorio de días sancionados, en caso de que existan (sólo para alumnos).
 - Servicio de reserva de aulas de estudio (sólo para alumnos).
 - Recordatorio de aulas reservadas (sólo para alumnos).
 - Recordatorio de libro reservado y posición.
 - Notificación de préstamos.
 - Recomendación de ejemplares nuevos o antiguos, acorde a los gustos y preferencias del usuario.
- Agente Cafetería/Comedor:
 - Notificación de localización de un amigo.
 - Notificación de reserva de mesas.
 - Recordatorio de reserva de mesas.
 - Recomendación de menú.
 - Recomendación de desayunos.
 - Recomendación de consumiciones.
 - Tablón de anuncios - cambio de turno (solo para alumnos).
 - Tablón de anuncios - cambio de libros (solo para alumnos).

- Agente Centro Cultural/Deportivo:
 - Servicio de reserva de entradas de eventos culturales/deportivos.
 - Servicio de reserva de instalaciones.
 - Recordatorio de entradas reservadas.
 - Recordatorio de instalaciones reservadas.
 - Recordatorio de días sin acudir al gimnasio/piscina (solo para alumnos).
 - Recomendación de eventos culturales / Recomendación de eventos deportivos.

Tecnologías utilizadas

Para la implementación del sistema MUSSUS se va a emplear el lenguaje de programación JAVA y, más concretamente, su arquitectura se basará en la plataforma JADE (Java Agent DEvelopment Framework) en la que los diferentes agentes del sistema podrán convivir de forma distribuida en los diferentes dispositivos, ya sean fijos en el caso de los servidores o móviles en el caso de los usuarios.

Para el almacenamiento de datos, se ha concluido que la mejor opción es el uso de archivos XML. Se trata de la mejor opción porque, como ya se ha visto en la gran mayoría de los sistemas mostrados en el estado del arte, se trata de un formato universal, de modo que cualquier tipo de terminal del que disponga el usuario podrá comprender la información de estos archivos y, por lo tanto, emplear el sistema. Esto no es muy importante para la simulación pero si en el caso de que se decidiese adaptar el sistema a un entorno real.

Para esta adaptación al entorno real, los usuarios podrían instalar el sistema en cualquier tipo de dispositivo móvil que tengan disponible: PC portátil, PDA, teléfono móvil 3G, etc. La única condición indispensable es que dicho dispositivo posea algún tipo de comunicación inalámbrica.

Las mejores opciones para la comunicación inalámbrica entre los distintos agentes del sistema, que será siempre una comunicación servidor-usuario y nunca usuario-usuario, serían las redes wifi o las basadas en tecnología Bluetooth. Para la localización de los usuarios podría pensarse en algún tipo de dispositivo como los RFIDs o los GPSs, pero en realidad también nos bastaría con la tecnología Bluetooth, ya que el agente servidor de cada zona podrá identificar a cada usuario mediante su nombre de dispositivo Bluetooth (que debería estar configurado con su NIA o PDI) una vez que este esté dentro de su radio de cobertura.

Escenario de uso

Un escenario sencillo de uso, pero que puede servir para comprender mejor el funcionamiento del sistema podría ser:

“David llega a la universidad y conecta el sistema en su teléfono móvil. Como la entrada está al lado la Secretaría lo primero que recibe es un mensaje indicándole que ha obtenido un 9.0 en la asignatura de “Ingeniería de Conocimiento”; el sistema le ofrece poner una reclamación pero como David está satisfecho con su nota decide cancelar el ofrecimiento. Además la Secretaría le recuerda que tiene pendiente la entrega de la práctica de “Programación Avanzada”. A continuación decide irse a desayunar a la Cafetería. Nada más entrar el sistema le ofrece unas patatas bravas que David acepta muy gustosamente, ya que no le entusiasma el dulce.

Quince minutos después, Eva llega a la universidad y conecta el sistema en su PDA. La primera zona por la que pasa todos los días es la de Secretaría y, al entrar en esta, el sistema le comunica que ha salido la nota del examen de “Redes de Ordenadores II” que estaba esperando y que su calificación es de 5.0. El sistema le ofrece la posibilidad de presentar una reclamación y Eva, al haber obtenido una calificación muy baja, la acepta y la envía con su queja correspondiente. Además el agente de la Secretaría le comunica que han encontrado una cartera roja que podría ser la que Eva había perdido hace unos días. Antes de ir a clase Eva acude a la Cafetería en busca de su amigo David y al entrar en dicha zona el agente correspondiente le indica automáticamente que este se encuentra ya allí y, a continuación, le recomienda desayunar un donut y un café con leche, aunque en esta ocasión Eva lo rechaza porque tiene prisa. Después de las clases, va a la Biblioteca a devolver un libro, cosa que le es avisada automáticamente nada más entrar en la Biblioteca. Además le llega la notificación de que ha entrado un nuevo libro de Laura Gallego, su autora favorita, que podría interesarle y Eva lo acepta y se lo lleva. Al finalizar el día Eva pasa por el Centro Cultural y Deportivo para recoger unas entradas para un partido de fútbol. Nada más entrar, el sistema le recuerda que lleva 12 días sin pasar por el gimnasio. A cambio de la regañina le recomienda un espectáculo de teatro para el viernes por la noche, que por supuesto acepta para ella y otras 3 personas, volviendo a casa con planes para el fin de semana.

Esa misma tarde, Francisco termina de impartir sus clases y decide pasar por Secretaría como cada tarde para comprobar su casillero. Como es nuevo en la universidad a menudo se desorienta, por lo que consulta el mapa para ver donde se encuentra. Cuando llega a la Secretaría se le comunica que tiene 5 entregas en su casillero, además de la reclamación que Eva le había presentado esa misma mañana. Además se le indica que un profesor al que estaba buscando se encuentra en su despacho en este momento.”

4. Roles, agentes y ontología de MUSSUS

A la hora de realizar un Sistema Multi-Agente, antes de ponerse a implementar los agentes, es necesario definir claramente dicho sistema. En primer lugar, es necesario explicar el problema a tratar, el dominio al que se aplica, cuales son los roles y servicios de cada agente y, sobre todo la asignación de responsabilidades. Tras haber explicado todo esto, nos queda lo más importante a la hora de crear un Sistema Multi-Agente que es la definición de su ontología.

Por lo tanto, en esta sección se hace una definición de los principales términos utilizados en los mensajes de este sistema, para ello se definirá un modelo de roles y una definición y representación de los principales conceptos del sistema.

Modelo de roles

Análisis de recursos

En primer lugar, debe realizarse un análisis de los recursos de los que dispondrá el sistema. Estos recursos serán:

- **mapa:** mapa en forma de cuadrícula en el cual se divide la universidad para delimitar las *Zonas* y en el cual se encuentran los *Usuarios*, pudiendo haber varios usuarios en una misma celda.
- **perfil:** datos asociados al usuario, tanto personales como referentes a sus gustos y deseos.
- **casillero:** buzón asociado a un profesor donde se le entregan los trabajos y otros documentos. El atributo *cantidad* indica el número de elementos que contiene el buzón.
- **asignatura:** materia en la que están matriculados varios alumnos y que es impartida por uno o varios profesores.
- **calificación:** está asociada a una *asignatura* y al *NIA* de un alumno. El atributo *nota* indica la calificación obtenida por el alumno en dicha asignatura.
- **reclamación:** petición del alumno de que le sea revisada una *nota* con la que no esta de acuerdo. Sus atributos serán la *asignatura* a la que se refiere la reclamación, el *NIA* del alumno que la ha puesto y el *texto* de esta, que es introducido por dicho alumno.
- **trabajo:** actividad o práctica que debe entregar un alumno. Estará asociado a una *asignatura* y tendrá una *fecha* y un *profesor* al cual se le debe entregar.
- **examen:** prueba que debe realizar un alumno para obtener sus calificaciones. Estará asociado a una *asignatura* y tendrá una *fecha* y un *lugar* de realización.
- **objetoPerdido:** elemento que un usuario ha extraviado. Se identificará mediante un nombre de *objeto* descriptivo.
- **ejemplar:** libro almacenado en la biblioteca y que puede ser consultado y tomado prestado por un usuario. Tendrá un *título* y también un código *isbn* para poder ser identificado unívocamente. En caso de que un usuario tome un ejemplar prestado, este tendrá también una *fecha de devolución* asociada.

- **díasSanción:** se produce cuando un usuario se retrasa en la devolución de un ejemplar. En ese caso se penalizará con un número de *días* sin poder retirar ejemplares.
- **reservaEjemplar:** se produce cuando un usuario desea un ejemplar que no está disponible y lo solicita para cuando lo esté. Sus atributos serán el *NIA* del alumno, el *ISBN* del ejemplar y la *posición* en la que se encuentra para recibir el ejemplar después de otros usuarios.
- **aulaEstudio:** en la biblioteca existen pequeñas aulas que pueden ser reservadas por los usuarios para estudiar o realizar trabajos en grupo. Estas aulas tendrán un número de *aula* asociado.
- **reservaAula:** cuando un usuario reserva un aula de estudio, esta reserva tendrá el *NIA* del alumno, el número de *aula* reservada, así como la *fecha* y *hora* de la reserva. Todas las reservas serán de una hora de duración.
- **menú:** conjunto de platos servidos en el comedor a medio día. Estos menús podrán ser recomendados a los usuarios y, para ello, tendrán un código de *menú* y una descripción para que el usuario sepa el contenido de dicho menú, que consistirá en el *primer* y *segundo* plato, el *postre* y las *propiedades* del menú.
- **desayuno:** conjunto de platos servidos por las mañanas. Estos desayunos podrán ser recomendados a los usuarios y, para ello, tendrán un código de *desayuno* y una descripción para que el usuario sepa el contenido de dicho desayuno, que consistirá en una *bebida*, una *comida* y las *propiedades* del desayuno.
- **consumición:** bebidas o comidas que son servidas en la cafetería durante todo el día. Estas consumiciones podrán ser recomendadas a los usuarios y para ello tendrán un *código*, un *nombre* y unas *propiedades* para que el usuario sepa el contenido tenga más información sobre dicha consumición.
- **mesa:** sitios en los que sentarse en la cafetería y el comedor. Tendrán un número de *mesa* asociado a cada una de ellas.
- **reservaMesa:** un usuario podrá reservar una mesa cuando lo desee. Esta reserva tendrá el *ID* del usuario, el número de *mesa*, y la *fecha* y *hora* de la reserva.
- **tablónAnuncios:** lugar donde los alumnos pueden publicar información relativa a:
 - **cambiosTurno:** un alumno podrá solicitar encontrar a otro alumno para un cambio de turno. Para ello será necesario conocer el *curso* y la *titulación* para los cuales se desea el cambio, el *turno_actual* que tiene el alumno y el *turno_deseado* al que quiere cambiar. En caso de encontrar a un alumno apto, el sistema devolverá el *NIA* del alumno encontrado.
 - **cambiosLibro:** un alumno podrá solicitar encontrar a otro alumno para un cambio de libros. Para ello será necesario conocer el *título* del libro para el que se desea el cambio y que *tipo* de petición está realizando el alumno (Oferta o Demanda). En caso de encontrar a un alumno apto, el sistema devolverá el *NIA* del alumno encontrado.

- **evento:** pasatiempo o espectáculo que se lleva a cabo en la Universidad o lugares asociados a ésta. Tendrán un *código* y un *nombre* de evento, un *tipo* (ya que podrá ser un evento Cultural o Deportivo), el *lugar* donde se realiza, una *fecha* y *hora* en la que se produce, una *descripción* y la *cantidad* de entradas que quedan disponibles.
- **reservaEvento:** un usuario podrá reservar entradas para un evento cultural o deportivo. Esta reserva tendrá el *ID* del usuario, el *código* y *tipo* del evento y, además, la *cantidad* de entradas deseadas para éste.
- **instalación:** zona del centro deportivo, como una pista de tenis, un campo de fútbol, etc. Estas se identificarán mediante un nombre de *instalación*.
- **reservaInstalación:** un usuario podrá reservar instalaciones para su uso personal. Esta reserva tendrá el *ID* del usuario, el *nombre* de la instalación reservada y la *fecha* y *hora* de la reserva. Todas las reservas serán de una hora de duración.
- **asistenciaGimnasio y asistenciaPiscina:** el sistema memorizará la asistencia de los alumnos a la piscina y el gimnasio, pudiendo así informarle de los *días* que lleva sin acudir.

Análisis de interfaces externas

El sistema tendrá una interfaz externa que es Usuario@. A través de esta los usuarios podrán ponerse en contacto con el sistema y para ello se comunicarán con los roles Profesor o Alumno, según sea su categoría.

Análisis de roles

Los roles identificados en el sistema son:

- Localizador: será el encargado de localizar a los usuarios en el mapa y de iniciar o terminar las comunicaciones con las zonas en las que se encuentren.
- Secretaría: será el rol encargado de llevar a cabo las tareas propias de la secretaría de la universidad, como gestionar la publicación de notas, los recordatorios de trabajos y exámenes o la localización de objetos perdidos.
- Biblioteca: será el rol encargado de llevar a cabo las tareas propias de la biblioteca de la universidad, como gestionar el préstamo de libros, los recordatorios de días de sanción o la reserva de aulas de estudio, así como la recomendación de ejemplares según los gustos del usuario.
- Cafetería: será el rol encargado de llevar a cabo las tareas propias de la cafetería de la universidad incluyendo la localización de amigos y la recomendación de desayunos y otras consumiciones según los gustos del usuario. Además facilitará los intercambios de libros y turnos para los alumnos mediante su tabón de anuncios.
- Comedor: será el rol encargado de llevar a cabo las tareas propias del comedor de la universidad incluyendo la reserva de mesas y la recomendación de menús según los gustos del usuario.
- Centro Cultural: será el rol encargado de llevar a cabo las tareas propias del centro cultural de la universidad, como la reserva de entradas para espectáculos o la recomendación de algún evento cultural según los gustos del usuario.
- Centro Deportivo: será el rol encargado de llevar a cabo las tareas propias del centro deportivo de la universidad, como la reserva de entradas para partidos, la reserva de instalaciones o la recomendación de algún evento deportivo según los gustos del usuario.
- Alumno: será el rol asignado a un usuario que recibe clases en la universidad y que podrá disfrutar de los servicios que proporciona el sistema.
- Profesor: será el rol asignado a un usuario que imparte clases en la universidad y que podrá disfrutar de los servicios que proporciona el sistema.

Especificación de roles

A continuación, se verá una especificación más concreta de los roles mencionados (donde **lep** serán permisos de lectura-escritura-propiedad).

Localizador
<p>Descripción:</p> <p>Este rol será el encargado de localizar a los usuarios en el mapa y conocerá los límites de cada zona. Además se encargará de iniciar o terminar las comunicaciones entre las zonas y los usuarios que se encuentren en ellas; para ello, deberá tener un registro de los usuarios y zonas existentes en el sistema en cada momento. También se encargará del apagado del sistema.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, DesconexiónUsuarios, DesconexiónZonas y DesconexiónSistema.</p> <p><u>Apagar.</u></p>
<p>Permisos:</p> <p>lep mapa</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - coordenadas: $0 < x, y < 9$

Tabla 2 - Rol Localizador

Secretaría
<p>Descripción:</p> <p>Este rol será el encargado de llevar a cabo las tareas propias de la secretaría de la universidad, como gestionar la publicación de notas, los casilleros, los recordatorios de trabajos y exámenes o la localización de personas u objetos perdidos.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónZonas, DesconexiónSistema, NotificaciónCasillero, NotificaciónReclamaciones, NotificaciónNotas, RecordatorioPlazosEntrega, RecordatorioExámen, NotificaciónProfesorLocalizado, NotificaciónAlumnoLocalizado y NotificaciónObjetoPerdido.</p> <p><u>ObtenerResultadosAlumno</u> y <u>ObtenerResultadosProfesor</u>.</p>
<p>Permisos:</p> <p>lep casillero</p> <p>lep asignatura</p> <p>lep calificación</p> <p>lep reclamación</p> <p>lep trabajo</p> <p>lep exámen</p> <p>lep objetoPerdido</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - fecha_trabajo >= fecha_actual - fecha_exámen >= fecha_actual

Tabla 3 - Rol Secretaria

Biblioteca
<p>Descripción:</p> <p>Este rol será el encargado de llevar a cabo las tareas propias de la biblioteca de la universidad, como gestionar el préstamo de libros, los recordatorios de días de sanción o la reserva de aulas de estudio, así como la recomendación de ejemplares según los gustos del usuario.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónZonas, DesconexiónSistema, RecordatorioEntregaEjemplar, RecordatorioDíasSanción, ReservarAulas, RecordatorioAulaReservada, RecordatorioLibroReservado, NotificaciónPréstamo y RecomendaciónEjemplar.</p> <p><u>ObtenerResultadosAlumno</u> y <u>ObtenerResultadosProfesor</u>.</p>
<p>Permisos:</p> <p>lep ejemplar</p> <p>lep díasSanción</p> <p>lep reservaEjemplar</p> <p>lep aulaEstudio</p> <p>lep reservaAula</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - fecha_préstamo > fecha_actual - fecha_aula >= fecha_actual - no se puede realizar un préstamo si díasSanción>0 - no se puede realizar una misma recomendación más de una vez.

Tabla 4 - Rol Biblioteca

Cafetería
<p>Descripción:</p> <p>Este rol será el encargado de llevar a cabo las tareas propias de la cafetería de la universidad incluyendo la localización de amigos y la recomendación de desayunos y otras consumiciones según los gustos del usuario. Además facilitará los intercambios de libros y turnos para los alumnos mediante su tablón de anuncios.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónZonas, DesconexiónSistema, NotificaciónAmigoLocalizado, RecomendaciónDesayunos, RecomendaciónConsumiciones, TablónCambioTurnos y TablónCambioLibros.</p> <p><u>ObtenerResultadosAlumno</u> y <u>ObtenerResultadosProfesor</u>.</p>
<p>Permisos:</p> <p>lep desayuno</p> <p>lep consumición</p> <p>lep tablónAnuncios</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - los desayunos se recomendarán hasta las 12:00. - las consumiciones se recomendarán a partir de las 16:00 o si no existe una recomendación mejor. - no se puede realizar una misma recomendación más de una vez. - el tablón no puede realizar un mismo intercambio a más de dos personas.

Tabla 5 - Rol Cafetería

Comedor
<p>Descripción:</p> <p>Este rol será el encargado de llevar a cabo las tareas propias del comedor de la universidad incluyendo la reserva de mesas y la recomendación de menús según los gustos del usuario.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónZonas, DesconexiónSistema, NotificaciónReservaMesa, RecordatorioReservaMesa y RecomendaciónMenús.</p> <p><u>ObtenerResultadosAlumno</u> y <u>ObtenerResultadosProfesor</u>.</p>
<p>Permisos:</p> <p>lep menú</p> <p>lep mesa</p> <p>lep reservaMesa</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - los menús se recomendarán desde las 12:00 hasta las 16:00. - no se puede realizar una misma recomendación más de una vez. - fecha_reserva >= fecha_actual - no puede reservarse un número de mesa inexistente.

Tabla 6 - Rol Comedor

Centro Cultural
<p>Descripción:</p> <p>Este rol será el encargado de llevar a cabo las tareas propias del centro cultural de la universidad, como la reserva de entradas para espectáculos o la recomendación de algún evento cultural según los gustos del usuario.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónZonas, DesconexiónSistema, ReservaEntradasEventos, RecordatorioEntradasReservadas y RecomendaciónEventos.</p> <p><u>ObtenerResultadosAlumno</u> y <u>ObtenerResultadosProfesor</u>.</p>
<p>Permisos:</p> <p>lep evento</p> <p>lep reservaEvento</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - no se puede realizar una misma recomendación más de una vez. - no se pueden reservar más entradas de las disponibles. - no puede reservarse un número de evento inexistente. - fecha_evento >= fecha_actual.

Tabla 7 - Rol Centro Cultural

Centro Deportivo
<p>Descripción:</p> <p>Este rol será el encargado de llevar a cabo las tareas propias del centro deportivo de la universidad, como la reserva de entradas para partidos, la reserva de instalaciones o la recomendación de algún evento deportivo según los gustos del usuario.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónZonas, DesconexiónSistema, ReservaEntradasEventos, ReservaInstalaciones, RecordatorioEntradasReservadas, RecordatorioInstalacionesReservadas, RecordatorioDíasAusencia y RecomendaciónEventos.</p> <p><u>ObtenerResultadosAlumno</u> y <u>ObtenerResultadosProfesor</u>.</p>
<p>Permisos:</p> <p>lep evento</p> <p>lep reservaEvento</p> <p>lep instalación</p> <p>lep reservaInstalación</p> <p>lep asistenciaGimnasio</p> <p>lep asistenciaPiscina</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - no se puede realizar una misma recomendación más de una vez. - no se pueden reservar más entradas de las disponibles. - no puede reservarse un número de instalación o de evento inexistente. - fecha_evento >= fecha_actual. - fecha_reserva_instalacion >= fecha_actual.

Tabla 8 - Rol Centro Deportivo

Alumno
<p>Descripción:</p> <p>Este rol será el asignado a un usuario que recibe clases en la universidad y que podrá disfrutar de los servicios que proporciona el sistema.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónSistema, NotificaciónNotas, RecordatorioPlazosEntrega, RecordatorioExámen, NotificaciónProfesorLocalizado, NotificaciónAlumnoLocalizado, NotificaciónObjetoPerdido, RecordatorioEntregaEjemplar, RecordatorioDíasSanción, ReservarAulas, RecordatorioAulaReservada, RecordatorioLibroReservado, NotificaciónPréstamo, RecomendaciónEjemplar, NotificaciónAmigoLocalizado, RecomendaciónDesayunos, RecomendaciónConsumiciones, TablónCambioTurnos, TablónCambioLibros, NotificaciónReservaMesa, RecordatorioReservaMesa, RecomendaciónMenús, ReservaEntradasEventos, RecordatorioEntradasReservadas, RecomendaciónEventos, ReservaInstalaciones, RecordatorioDíasAusencia y RecordatorioInstalacionesReservadas.</p>
<p>Permisos:</p> <p>lep perfil</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - en todo el perfil: fecha_petición >= fecha_actual.

Tabla 9 - Rol Alumno

Profesor
<p>Descripción:</p> <p>Este rol será el asignado a un usuario que imparte clases en la universidad y que podrá disfrutar de los servicios que proporciona el sistema.</p>
<p>Protocolos y actividades:</p> <p>MovimientoUsuarios, EnvíoInformación, DesconexiónUsuarios, DesconexiónZonas, DesconexiónSistema, NotificaciónCasillero, NotificaciónReclamaciones, NotificaciónProfesorLocalizado, NotificaciónAlumnoLocalizado, NotificaciónObjetoPerdido, RecordatorioEntregaEjemplar, RecordatorioLibroReservado, NotificaciónPréstamo, RecomendaciónEjemplar, NotificaciónAmigoLocalizado, RecomendaciónDesayunos, RecomendaciónConsumiciones, NotificaciónReservaMesa, RecordatorioReservaMesa, RecomendaciónMenús, ReservaEntradasEventos, RecordatorioEntradasReservadas, RecomendaciónEventos, ReservaInstalaciones y RecordatorioInstalacionesReservadas.</p>
<p>Permisos:</p> <p>lep perfil</p>
<p>Responsabilidades:</p> <ul style="list-style-type: none"> • de seguridad: <ul style="list-style-type: none"> - en todo el perfil: fecha_petición >= fecha_actual.

Tabla 10 - Rol Profesor

Modelo de Agentes

De acuerdo a los roles especificados anteriormente, se han identificado los agentes del sistema que se corresponderán en su mayoría con un solo rol. Solo en el caso del Agente Cafetería y el Agente CentroCyD se han englobado dos roles en un solo agente. Se ha tomado esta decisión debido a que estos roles operan sobre la misma zona, siendo además sus funciones muy similares o relacionadas. Como puede verse también, la única función del Agente Universidad es hacer de localizador de los agentes del sistema, siendo todas las demás funciones competencia de las zonas a las que les corresponda.

El sistema contará por tanto con un Agente Universidad, un agente de cada una de las zonas y un número indeterminado de agentes Alumnos y Profesores.

En el siguiente gráfico se muestra la correspondencia entre los roles y los agentes del sistema.

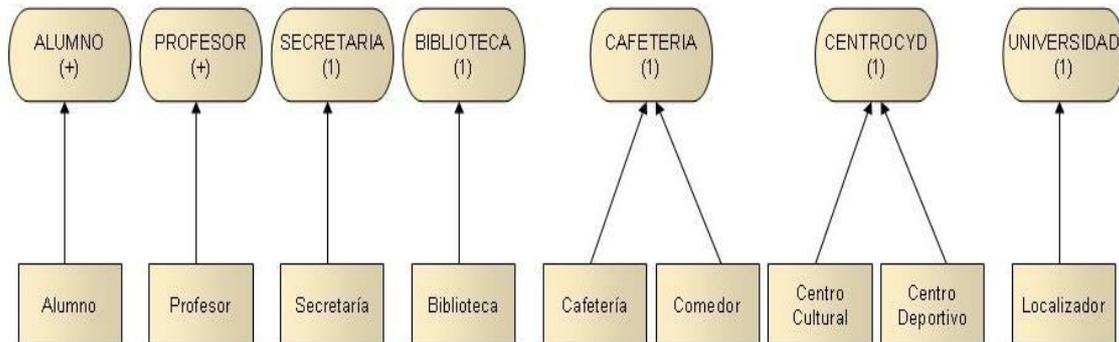


Ilustración 6 - Correspondencia entre roles y agentes

Agente Biblioteca

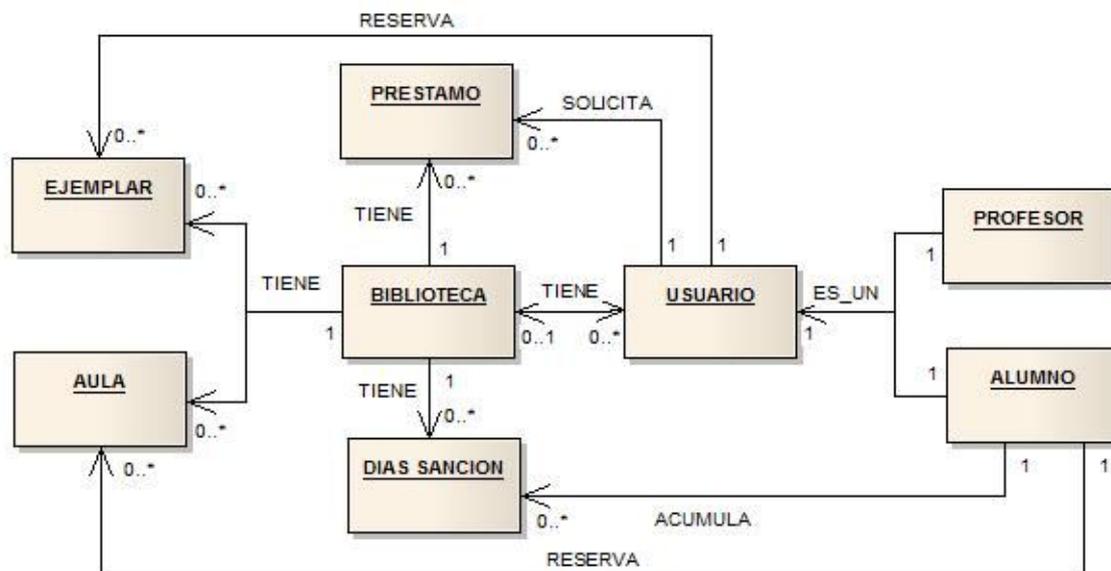


Ilustración 9 - Conceptos del agente biblioteca

Agente Cafetería/Comedor

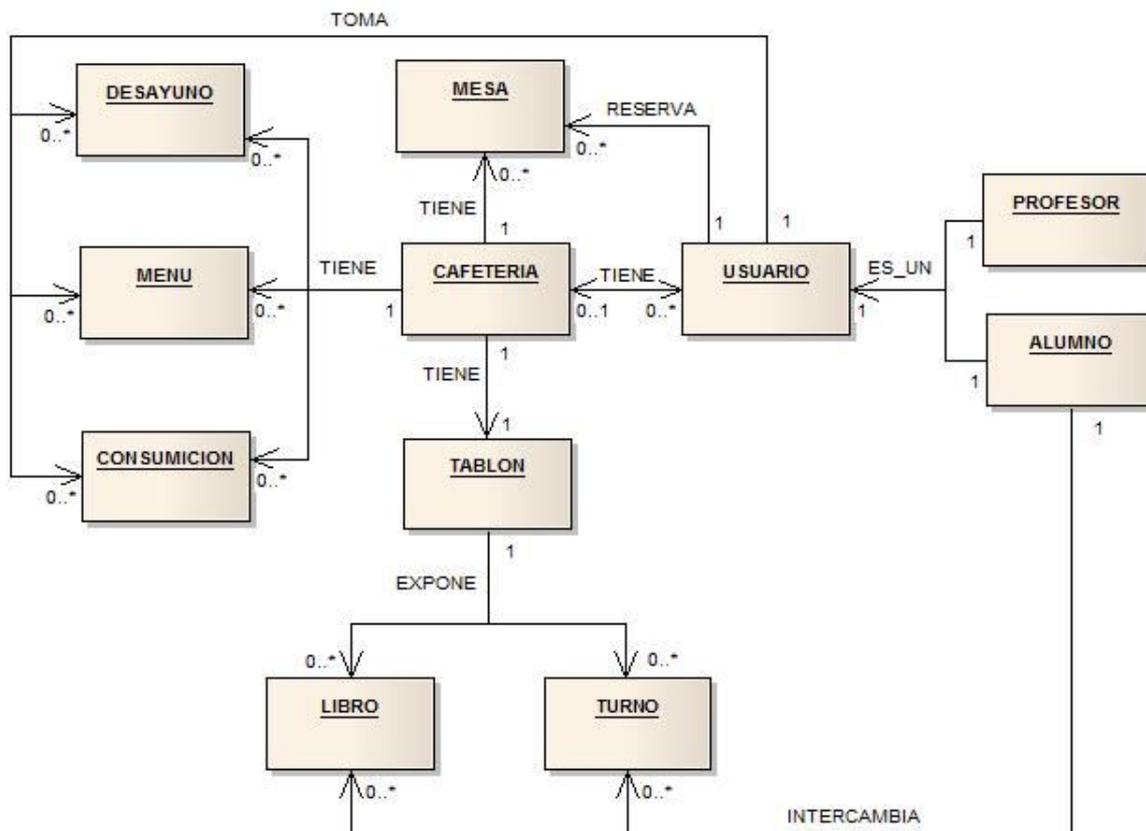


Ilustración 10 - Conceptos del agente cafetería/comedor

Agente Centro Cultural/Deportivo

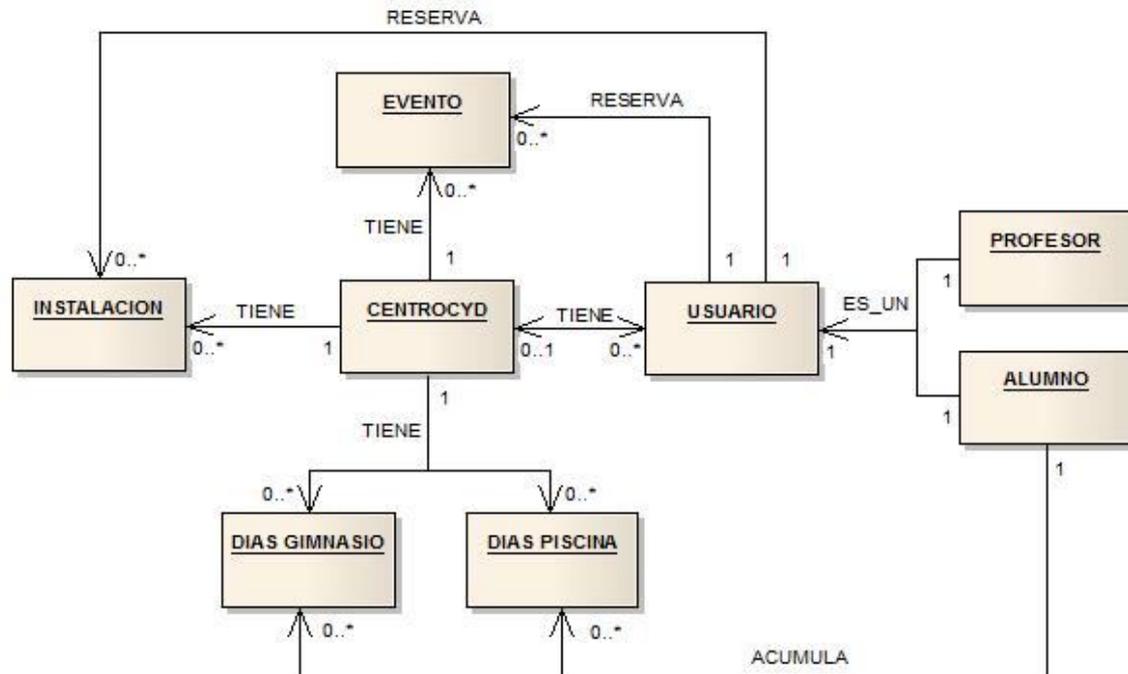


Ilustración 11 - Conceptos del agente centro cultural/deportivo

Como ya se ha comentado anteriormente, estos agentes serán implementados en lenguaje JAVA basándose en la plataforma JADE. En el **Anexo A** de este documento pueden verse los diagramas de clases correspondientes al sistema MUSSUS.

5. Protocolos de MUSSUS

En esta sección se mostrarán los protocolos de intercambio de mensajes entre los usuarios y los servidores del sistema. Este modelo de protocolos se mostrará dividido según las funcionalidades de cada servidor y por cada una de estas funcionalidades se mostrarán el formato de los mensajes intercambiados en su orden correspondiente de envío.

Protocolos Generales

Independientemente del tipo de usuario o de zona, existe un protocolo general para la detección de la entrada/salida de los usuarios en las zonas, así como para el intercambio de datos entre ambos.

Movimiento de usuarios

Protocolo: MovimientoUsuarios

Cuando un usuario cambia de posición envía un mensaje con sus nuevas coordenadas al Agente Universidad, cuya funcionalidad básica es la detección y ubicación de los usuarios. Además este agente se encarga de poner en contacto a dichos usuarios, con las zonas en las que se encuentren.

Este Agente Universidad dispone de un mapa en el que tiene ubicados a todos los usuarios, de este modo, cuando un usuario envía su posición, la universidad sabe automáticamente si ha cambiado o no de zona. En caso no cambiar, únicamente anota su nueva posición. Por el contrario, si el usuario cambia de zona, pueden darse dos casos:

- El usuario pasa de estar en el pasillo a entrar en una zona: en este caso, la universidad envía al usuario el mensaje de nueva zona y a la zona el mensaje de usuario presente. Dicha zona almacenará al nuevo usuario en su vector de usuarios. Por otro lado, el usuario procederá a enviar su perfil de acuerdo al protocolo *EnvíoInformación* que se muestra en el siguiente apartado.
- El usuario pasa de estar en una zona a salir al pasillo: en este caso, la universidad envía al usuario el mensaje de nueva zona (pasillo) y a la zona el mensaje de usuario ausente. Dicha zona eliminará al usuario de su vector de usuario así como su perfil.

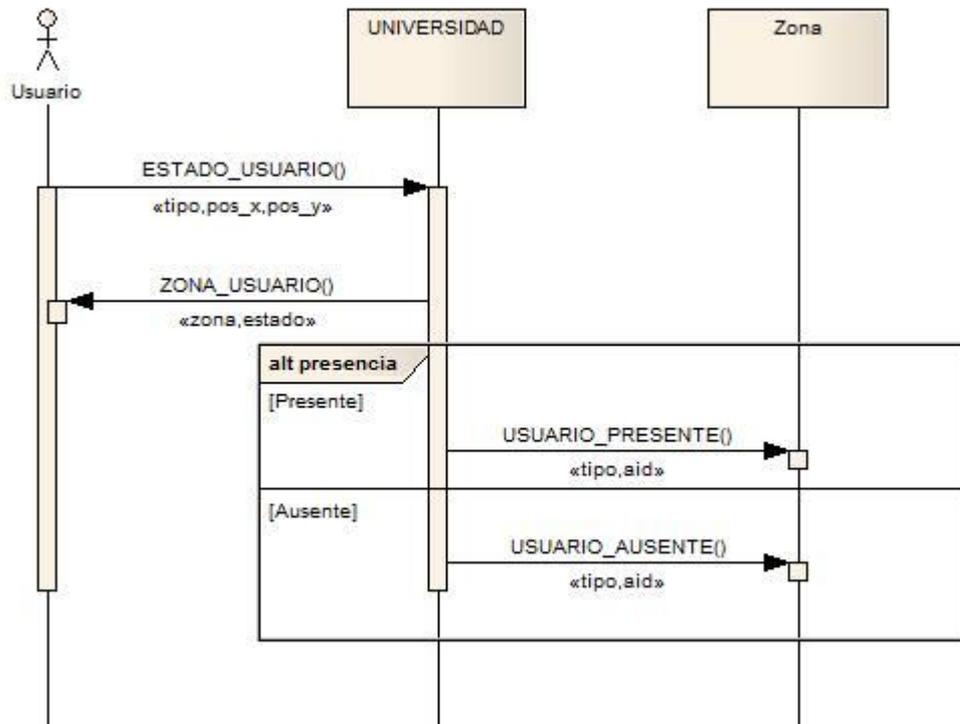


Ilustración 12 - Protocolo MovimientoUsuarios

ESTADO_USUARIO	
Descripción	Indica la posición en coordenadas de un usuario.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> a la Universidad de la posición del Usuario.
Emisor	USUARIO
Receptor	UNIVERSIDAD
ARGUMENTOS	
Argumento	Descripción
TIPO	Tipo de usuario: ALUMNO/PROFESOR.
POS_X	Coordenada X.
POS_Y	Coordenada Y.

Tabla 11 - Protocolo MovimientoUsuarios (Mensaje 1)

ZONA_USUARIO	
Descripción	Indica que el usuario ha cambiado de zona.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario de que ha cambiado de zona.
Emisor	UNIVERSIDAD
Receptor	USUARIO
ARGUMENTOS	
Argumento	Descripción
ZONA	Zona en la que ha entrado el usuario: SECRETARIA / CAFETERIA / BIBLIOTECA / CENTROCYD / PASILLO.
ESTADO	Indica si la zona está o no activa: OK / KO.

Tabla 12 - Protocolo MovimientoUsuarios (Mensaje 2)

USUARIO_PRESENTE	
Descripción	Indica que un usuario ha entrado en la zona.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> a la Zona de la presencia del Usuario.
Emisor	UNIVERSIDAD
Receptor	ZONA
ARGUMENTOS	
Argumento	Descripción
TIPO	Tipo de usuario: ALUMNO/PROFESOR.
AID	Id. del agente, que servirá para poder comunicarse con él.

Tabla 13 - Protocolo MovimientoUsuarios (Mensaje 3)

USUARIO_AUSENTE	
Descripción	Indica que un usuario ha salido de la zona.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> a la Zona de la ausencia del Usuario.
Emisor	UNIVERSIDAD
Receptor	ZONA
ARGUMENTOS	
Argumento	Descripción
TIPO	Tipo de usuario: ALUMNO/PROFESOR.
AID	Id. del agente, que servía para poder comunicarse con él.

Tabla 14 - Protocolo MovimientoUsuarios (Mensaje 4)

Envío de información

Protocolo: EnvíoInformación.

Cuando un usuario entra en una nueva zona (y por tanto recibe el mensaje ZONA_USUARIO descrito anteriormente) se activa el protocolo de envío de información, mediante el cual el usuario enviará a la zona en cuestión toda la información que esta necesite. Para ello, primero el usuario enviará su perfil genérico y, a continuación, la zona irá solicitando de forma secuencial los datos concretos que precise y el usuario se los enviará. La zona no solicitará nuevos datos hasta que no reciba los que ha pedido anteriormente.

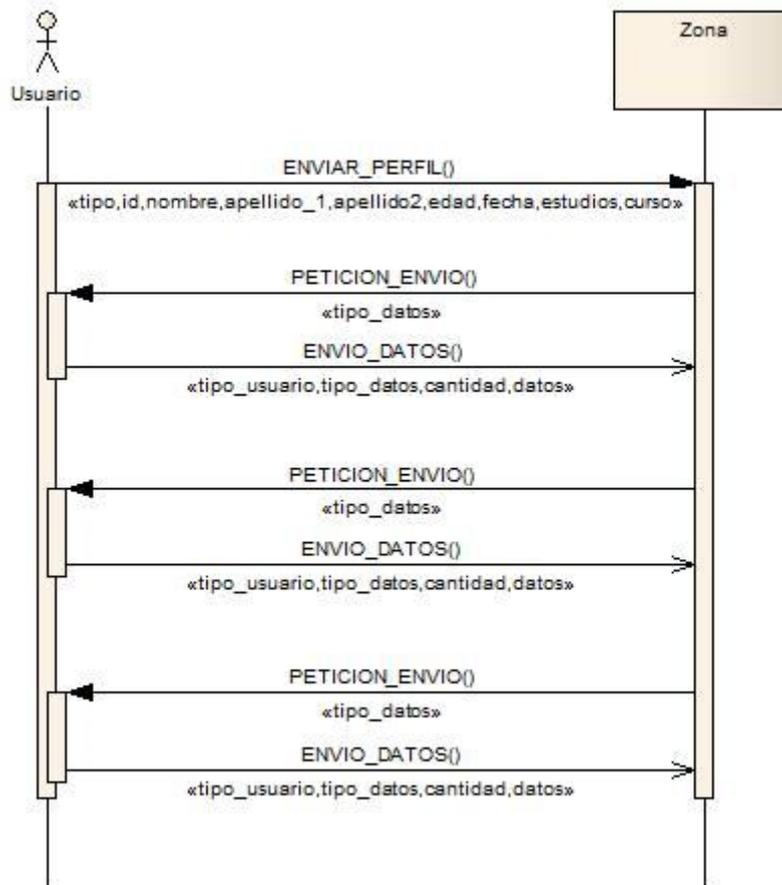


Ilustración 13 - Protocolo EnvíoInformación

ENVIAR_PERFIL	
Descripción	Mensaje con los datos de perfil del usuario.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para enviar un <i>informe</i> con el perfil del Usuario.
Emisor	USUARIO
Receptor	ZONA
ARGUMENTOS	
Argumento	Descripción
TIPO	Tipo de usuario: ALUMNO/PROFESOR.
ID	NIA/PDI del usuario.
NOMBRE	Nombre del usuario.
APELLIDO_1	Primero apellido del usuario.
APELLIDO_2	Segundo apellido del usuario.
EDAD	Edad del usuario.
FECHA	Fecha de nacimiento del usuario.
ESTUDIOS	Estudios cursados o impartidos por el usuario, según sea alumno/profesor
CURSO	Curso en el que está matriculado. Este campo será 0 en caso de tratarse de un profesor.

Tabla 15 - Protocolo EnvíoInformación (Mensaje 1)

PETICION_ENVIO	
Descripción	Mensaje de solicitud de envío de datos.
Performative	REQUEST La performative utilizada es REQUEST porque este mensaje sirve para <i>solicitar</i> el envío de unos datos concretos.
Emisor	ZONA
Receptor	USUARIO
ARGUMENTOS	
Argumento	Descripción
TIPO_DATOS	Tipo de datos solicitados: ASIGNATURAS, AMIGOS, etc.

Tabla 16 - Protocolo EnvíoInformación (Mensaje 2)

ENVIO_DATOS	
Descripción	Mensaje de envío de datos solicitados.
Performative	INFORM-REF La performative utilizada es INFORM-REF porque este mensaje sirve para enviar un <i>informe</i> con unos datos concretos previamente solicitados.
Emisor	USUARIO
Receptor	ZONA
ARGUMENTOS	
Argumento	Descripción
TIPO_USUARIO	Tipo de usuario: ALUMNO/PROFESOR.
TIPO_DATOS	Tipo de datos: ASIGNATURAS, AMIGOS, etc.
CANTIDAD	Número de datos enviados.
DATOS	Datos solicitados, separados por un espacio en blanco.

Tabla 17 - Protocolo EnvíoInformación (Mensaje 3)

Una vez que la zona tiene toda la información que necesita, se pone en marcha el algoritmo propio de esta para el tipo de usuario recibido. Los mensajes que se envíen entre ambos a partir de entonces serán los definidos en los protocolos concretos de cada zona, que se muestran más adelante.

Desconexión de usuarios

Protocolo: DesconexiónUsuarios

Cuando un usuario pulsa el botón para salir del sistema, automáticamente envía un mensaje al Agente Universidad indicándole que está desconectándose. Dicho Agente Universidad, al recibir el mensaje de desconexión del usuario, comprueba si este se encuentra dentro de una zona activa y, en caso afirmativo, envía un mensaje al Agente Zona en cuestión indicándole la desconexión del usuario. Finalmente, el Agente Universidad elimina al usuario del sistema.

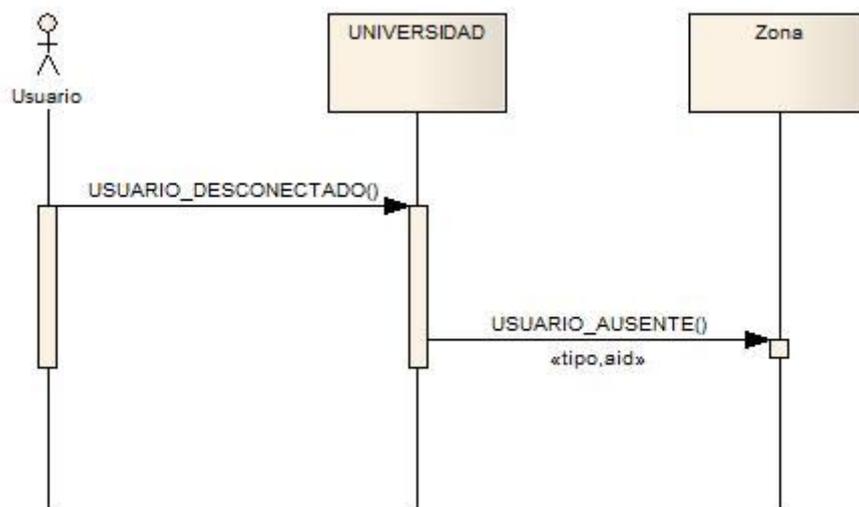


Ilustración 14 - Protocolo DesconexiónUsuarios

USUARIO_DESCONECTADO	
Descripción	Indica que el usuario se ha desconecta del sistema.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> a la Universidad de la desconexión del Usuario.
Emisor	USUARIO
Receptor	UNIVERSIDAD
ARGUMENTOS	
Argumento	Descripción

Tabla 18 - Protocolo DesconexiónUsuarios (Mensaje 1)

USUARIO_AUSENTE	
Descripción	Indica que un usuario ha salido de la zona.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> a la Zona de la ausencia del Usuario.
Emisor	UNIVERSIDAD
Receptor	ZONA
ARGUMENTOS	
Argumento	Descripción
TIPO	Tipo de usuario: USUARIO.
AID	Identificador del agente, que servía para poder comunicarse con él.

Tabla 19 - Protocolo DesconexiónUsuarios (Mensaje 2)

Desconexión de zonas

Protocolo: DesconexiónZonas

Cuando una zona se desconecta del sistema, automáticamente envía un mensaje al Agente Universidad indicándole que está desconectándose del sistema. Dicho Agente Universidad, al recibir el mensaje elimina a la zona del sistema.

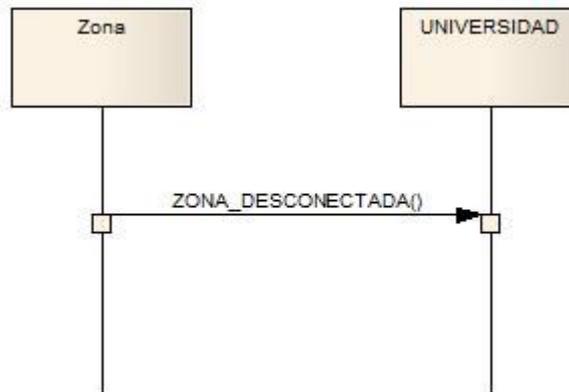


Ilustración 15 - Protocolo DesconexiónZonas

ZONA_DESCONECTADA	
Descripción	Indica que la zona se ha desconectado del sistema.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> a la Universidad de la desconexión de la Zona.
Emisor	ZONA
Receptor	UNIVERSIDAD
ARGUMENTOS	
Argumento	Descripción

Tabla 20 - Protocolo DesconexiónZonas (Mensaje 1)

Desconexión del sistema

Protocolo: DesconexiónSistema.

Cuando se pulsa el botón de Cerrar Aplicación, el Agente Universidad envía automáticamente el mensaje de desconexión a todas las zonas y usuarios conectados para indicarles que copien sus datos y después se desconecten también.

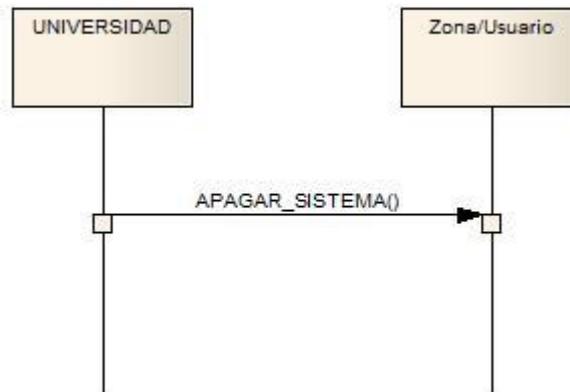


Ilustración 16 - Protocolo DesconexiónSistema

APAGAR_SISTEMA	
Descripción	Indica que el sistema se esta apagando.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> del apagado del sistema.
Emisor	UNIVERSIDAD
Receptor	ZONA/USUARIO
ARGUMENTOS	
Argumento	Descripción

Tabla 21 - Protocolo DesconexiónSistema (Mensaje 1)

Protocolos del Agente Secretaría

Notificación de entregas en el casillero (sólo para profesores)

Protocolo: NotificaciónCasillero.

Cuando un profesor entra en la zona de Secretaría, si tiene alguna entrega en su casillero, recibirá el siguiente mensaje del Agente Secretaría donde se le indicará cuantas entregas tiene.

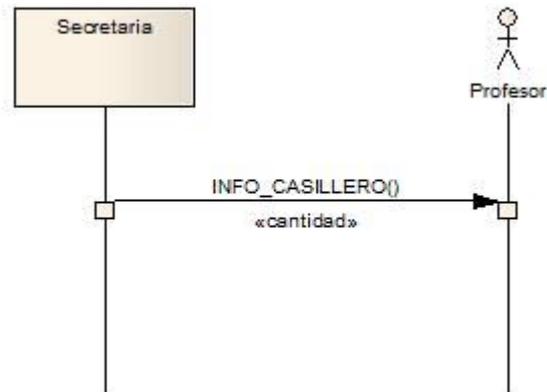


Ilustración 17 - Protocolo NotificaciónCasillero

INFO_CASILLERO	
Descripción	Indica al profesor que tiene entregas en su casillero.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al profesor de que tiene entregas.
Emisor	SECRETARÍA
Receptor	PROFESOR
ARGUMENTOS	
Argumento	Descripción
CANTIDAD	Número de entregas que el profesor tiene en su casillero.

Tabla 22 - Protocolo NotificaciónCasillero (Mensaje 1)

Notificación de reclamaciones (sólo para profesores)

Protocolo: NotificaciónReclamaciones.

Cuando un profesor entra en la zona de Secretaría, si tiene alguna reclamación pendiente, recibirá el siguiente mensaje del Agente Secretaría donde se le indicará de que reclamación se trata.

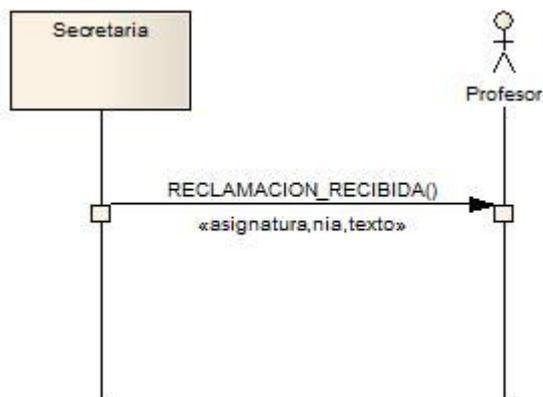


Ilustración 18 - Protocolo NotificaciónReclamaciones

RECLAMACION_RECIBIDA	
Descripción	Indica al profesor que tiene una reclamación.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al profesor de que tiene una reclamación.
Emisor	SECRETARÍA
Receptor	PROFESOR
ARGUMENTOS	
Argumento	Descripción
ASIGNATURA	Asignatura a la cual esta asociada la reclamación.
NIA	NIA del alumno que ha realizado la reclamación.
TEXTO	Texto de la reclamación.

Tabla 23 - Protocolo NotificaciónReclamaciones (Mensaje 1)

Notificación de publicación de notas / Posibilidad de escribir una reclamación a una nota (sólo para alumnos)

Protocolo: NotificaciónNotas.

Cuando un alumno entra en la zona de Secretaría, si ha salido una nota que estaba esperando, se producirá la siguiente secuencia de mensajes mediante la cual el Agente Secretaría le indica la nota y le ofrece la posibilidad de reclamar. El alumno contestará con el mensaje de aceptación, o nada en caso de rechazo.

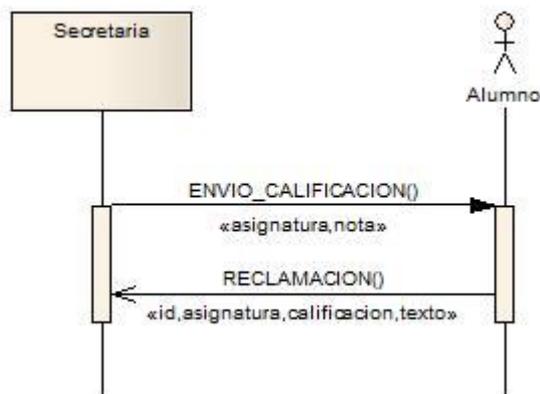


Ilustración 19 - Protocolo NotificaciónNotas

ENVIO_CALIFICACION	
Descripción	Indica al alumno que ha salido una de las notas que esperaba y le pregunta si desea reclamarla.
Performative	PROPOSE La performative utilizada es PROPOSE porque este mensaje sirve para enviar una calificación y <i>proponer</i> al usuario si desea reclamarla.
Emisor	SECRETARÍA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
ASIGNATURA	Nombre de la asignatura.
NOTA	Calificación obtenida en la asignatura.

Tabla 24 - Protocolo NotificaciónNotas (Mensaje 1)

RECLAMACION	
Descripción	Respuesta afirmativa del usuario a si desea reclamar el examen.
Performative	ACCEPT-PROPOSAL La performative utilizada es ACCEPT-PROPOSAL porque este mensaje sirve para <i>aceptar una propuesta</i> de reclamación.
Emisor	ALUMNO
Receptor	SECRETARÍA
ARGUMENTOS	
Argumento	Descripción
ID	NIA del alumno.
ASIGNATURA	Nombre de la asignatura.
CALIFICACIÓN	Calificación obtenida.
TEXTO	Texto de la reclamación.

Tabla 25 - Protocolo NotificaciónNotas (Mensaje 2)

Recordatorio de plazos de entrega de trabajos/prácticas (sólo para alumnos)

Protocolo: RecordatorioPlazosEntrega.

Cuando un alumno entra en la zona de Secretaría, si tiene algún trabajo pendiente de entregar, recibirá el siguiente mensaje del Agente Secretaría donde se le indica dicho trabajo, la fecha de entrega y el modo de la misma (ya sea en el casillero, por email, etc.).

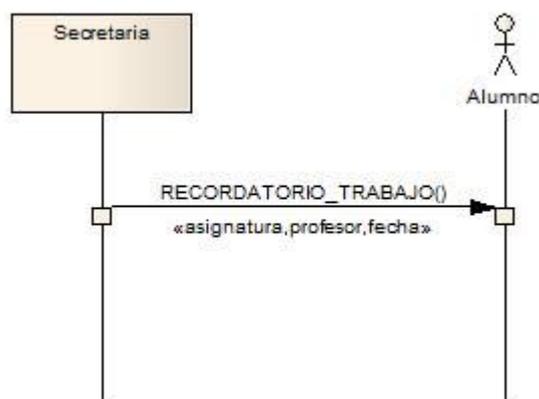


Ilustración 20 - Protocolo RecordatorioPlazosEntrega

RECORDATORIO_TRABAJO	
Descripción	Recuerda al alumno la fecha de entrega de algún trabajo pendiente.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Alumno sobre un trabajo pendiente.
Emisor	SECRETARÍA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
ASIGNATURA	Nombre de la asignatura.
PROFESOR	Nombre del profesor al que se le entrega.
FECHA	Fecha de la entrega.

Tabla 26 - Protocolo RecordatorioPlazosEntrega (Mensaje 1)

Recordatorio de fechas y lugares de exámenes (sólo para alumnos)

Protocolo: RecordatorioExámen.

Cuando un alumno entra en la zona de Secretaría, si tiene algún examen en la semana actual, recibirá el siguiente mensaje del Agente Secretaría donde se le indica dicho examen, la fecha de este y el lugar donde se realizará.

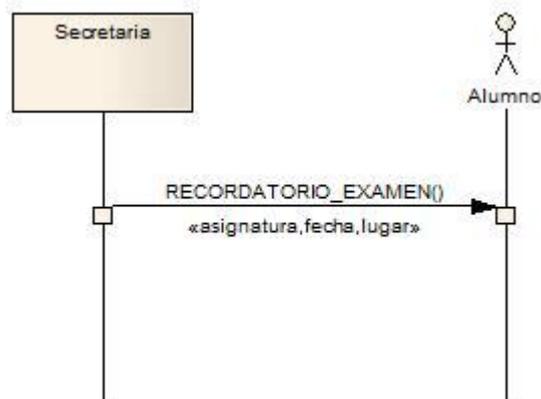


Ilustración 21 - Protocolo RecordatorioExámen

RECORDATORIO_EXAMEN	
Descripción	Recuerda al alumno la fecha de un examen pendiente.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Alumno sobre un examen pendiente.
Emisor	SECRETARÍA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
ASIGNATURA	Nombre de la asignatura.
FECHA	Fecha del examen.
LUGAR	Lugar donde se realizará el examen.

Tabla 27 - Protocolo RecordatorioExámen (Mensaje 1)

Notificación de localización de un profesor al que se está buscando

Protocolo: NotificaciónProfesorLocalizado.

Cuando un usuario entra en la zona de Secretaría, si estaba buscando a un profesor recibirá el siguiente mensaje del Agente Secretaría donde se le indica que profesor ha sido localizado y donde.

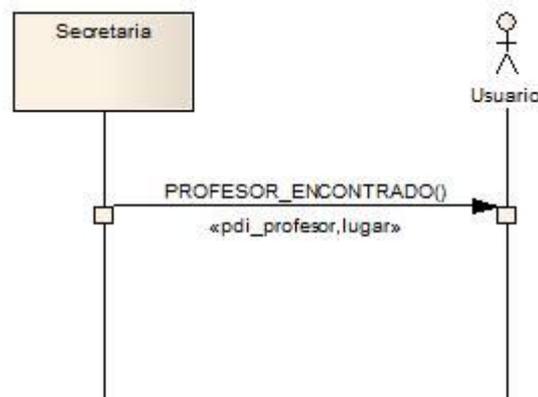


Ilustración 22 - Protocolo NotificaciónProfesorLocalizado

PROFESOR_ENCONTRADO	
Descripción	Indica al alumno o profesor que se ha localizado a un profesor al que estaba buscando.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre un profesor que ha sido encontrado.
Emisor	SECRETARÍA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
PDI_PROFESOR	PDI del profesor encontrado.
LUGAR	Lugar donde ha sido localizado el profesor.

Tabla 28 - Protocolo NotificaciónProfesorLocalizado (Mensaje 1)

Notificación de localización de un alumno al que se está buscando

Protocolo: NotificaciónAlumnoLocalizado.

Cuando un usuario entra en la zona de Secretaría, si estaba buscando a un alumno que se encuentra presente en ese momento, recibirá el siguiente mensaje del Agente Secretaría donde se le indica que el alumno ha sido localizado.

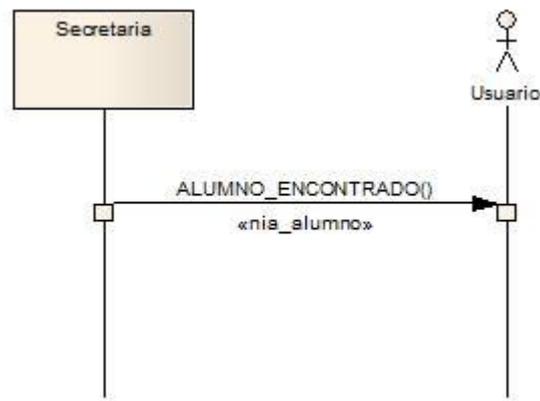


Ilustración 23 - Protocolo NotificaciónAlumnoLocalizado

ALUMNO_ENCONTRADO	
Descripción	Indica al alumno o profesor que se ha localizado a un alumno al que estaba buscando.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre un alumno que ha sido encontrado.
Emisor	SECRETARÍA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
NIA_ALUMNO	NIA del alumno localizado.

Tabla 29 - Protocolo NotificaciónAlumnoLocalizado (Mensaje 1)

Notificación de que un objeto perdido ha sido encontrado

Protocolo: NotificaciónObjetoPerdido.

Cuando un usuario entra en la zona de Secretaría, si está buscando un objeto que había perdido y este ha sido encontrado, recibirá el siguiente mensaje del Agente Secretaría donde se le indica que objeto ha sido localizado.



Ilustración 24 - Protocolo NotificaciónObjetoPerdido

OBJETO_ENCONTRADO	
Descripción	Indica al alumno o profesor que se ha localizado un objeto que había perdido.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre un objeto que ha sido encontrado.
Emisor	SECRETARÍA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
OBJETO	Nombre del objeto perdido.

Tabla 30 - Protocolo Notificación Objeto Perdido (Mensaje 1)

Protocolos del Agente Biblioteca

Recordatorio de plazos de entrega de ejemplares (sólo para alumnos)

Protocolo: RecordatorioEntregaEjemplar.

Cuando un alumno entra en la zona de Biblioteca, si tiene algún ejemplar en préstamo, recibirá el siguiente mensaje del Agente Biblioteca donde se le indica la fecha en que debe devolverlo.

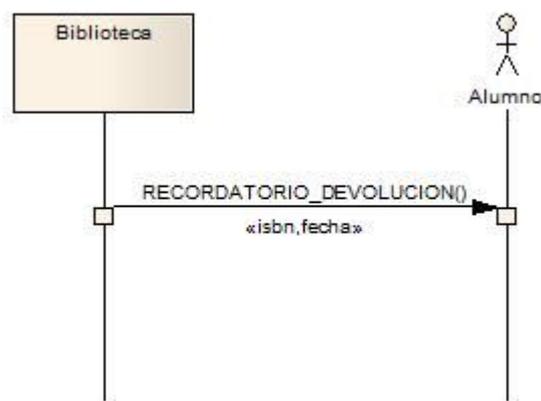


Ilustración 25 - Protocolo RecordatorioEntregaEjemplar

RECORDATORIO_DEVOLUCION	
Descripción	Recuerda al alumno la fecha de entrega de algún ejemplar de la biblioteca pendiente.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Alumno sobre un ejemplar pendiente.
Emisor	BIBLIOTECA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
ISBN	ISBN del ejemplar que debe ser entregado.
FECHA	Fecha de la entrega.

Tabla 31 - Protocolo RecordatorioEntregaEjemplar (Mensaje 1)

Recordatorio de días sancionados, en caso de que existan (sólo para alumnos)

Protocolo: RecordatorioDíasSanción.

Cuando un alumno entra en la zona de Biblioteca, si tiene días de sanción, recibirá el siguiente mensaje del Agente Biblioteca donde se le indica cuantos días está sancionado.

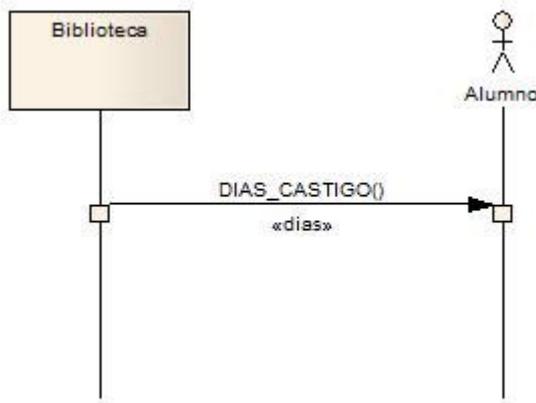


Ilustración 26 - Protocolo RecordatorioDíasSanción

DIAS_CASTIGO	
Descripción	Recuerda al alumno o profesor los días de sanción que tiene.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Alumno de los días que tiene de sanción.
Emisor	BIBLIOTECA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
DÍAS	Número de días que el alumno tiene de sanción.

Tabla 32 - Protocolo RecordatorioDíasSanción (Mensaje 1)

Servicio de reserva de aulas de estudio (sólo para alumnos)

Protocolo: ReservaAulas.

Cuando un alumno desea reservar un aula de estudio deberá hacerlo en la zona de Biblioteca. A partir de la fecha y hora a la cual se desea reservar dicha aula, el Agente Biblioteca envía un mensaje indicando si hay o no un aula disponible y, si la hay, le indicará el número de aula asignada.

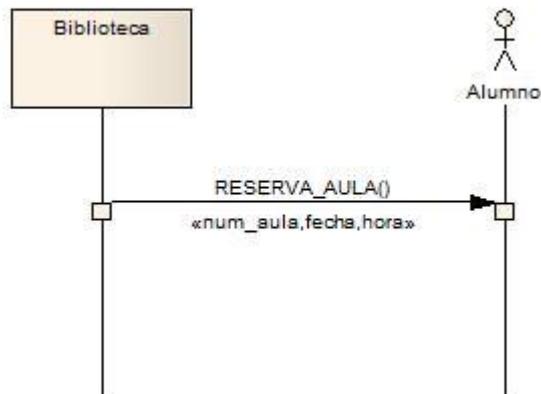


Ilustración 27 - Protocolo ReservaAulas

RESERVA_AULA	
Descripción	Mensaje del Agente Biblioteca con la aceptación o no de la reserva y el aula correspondiente.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> al Alumno de si le ha sido o no reservada un aula.
Emisor	BIBLIOTECA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
NUM_AULA	Número de aula reservada (este valor será 0 para el caso de aula no disponible).
FECHA	Fecha de la reserva.
HORA	Hora de la reserva.

Tabla 33 - Protocolo ReservaAulas (Mensaje 1)

Recordatorio de aulas reservadas (sólo para alumnos)

Protocolo: RecordatorioAulaReservada.

Cuando un usuario entra en la zona de Biblioteca, si tiene un aula de estudio reservada, recibirá el siguiente mensaje del Agente Biblioteca donde se le indica que aula es y en que fecha y hora esta reservada para él.

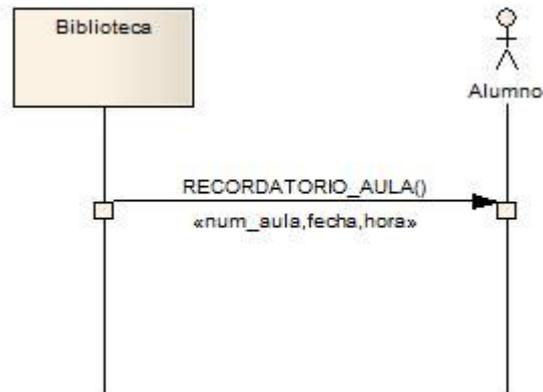


Ilustración 28 - Protocolo RecordatorioAulaReservada

RECORDATORIO_AULA	
Descripción	Recuerda al alumno que tiene un aula reservada.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Alumno sobre un aula reservada.
Emisor	BIBLIOTECA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
NUM_AULA	Número de aula reservada.
FECHA	Fecha de la reserva.
HORA	Hora de la reserva.

Tabla 34 - Protocolo RecordatorioAulaReservada (Mensaje 1)

Recordatorio de libro reservado y posición

Protocolo: RecordatorioLibroReservado.

Cuando un usuario entra en la zona de Biblioteca, si tiene un ejemplar reservado se le notifica con la posición de reserva en la que se encuentra en ese momento.

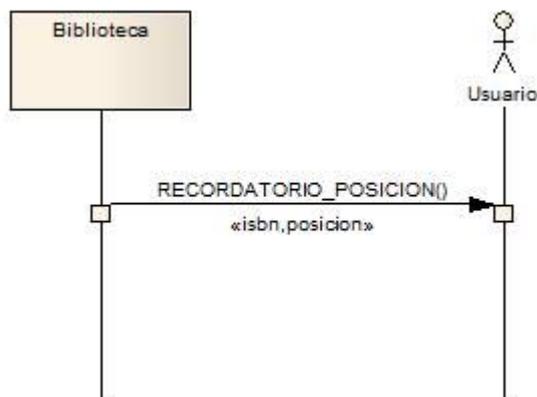


Ilustración 29 - Protocolo RecordatorioLibroReservado

RECORDATORIO_POSICION	
Descripción	Recordatorio de ejemplar reservado y la correspondiente posición en la lista de reserva.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre un libro reservado.
Emisor	BIBLIOTECA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
ISBN	Código ISBN del ejemplar reservado.
POSICION	Posición en la lista de reserva.

Tabla 35 - Protocolo RecordatorioLibroReservado (Mensaje 1)

Notificación de préstamos

Protocolo: NotificaciónPréstamo.

Cuando un usuario entra en la zona de Biblioteca, si ha solicitado algún libro o tenía alguna reserva pendiente que ya puede recoger recibe un mensaje indicando el estado de dicho préstamo o solicitud.



Ilustración 30 - Protocolo NotificaciónPréstamo

LIBRO_PRESTADO	
Descripción	Notificación del estado de un préstamo o solicitud.
Performative	INFORM-IF La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre si un ejemplar le ha sido o no prestado.
Emisor	BIBLIOTECA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
ESTADO	Estado del préstamo, pudiendo ser: <ul style="list-style-type: none"> ▪ PRESTADO: indica que un ejemplar solicitado ha sido prestado con éxito. ▪ PRESTADORES: indica que un ejemplar reservado ha sido prestado con éxito. ▪ NO_DISPONIBLE: indica que el ejemplar solicitado no está disponible en este momento. ▪ INEXISTENTE: indica que el ejemplar solicitado no existe.
ISBN	Código ISBN del ejemplar.
TITULO	Título del ejemplar.
FECHA	Fecha en la que debe devolverse el ejemplar en caso de préstamo.

Tabla 36 - Protocolo NotificaciónPréstamo (Mensaje 1)

Recomendación de ejemplares nuevos o antiguos, acorde a los gustos y preferencias del usuario

Protocolo: RecomendaciónEjemplar.

Cuando un usuario entra en la zona de Biblioteca, si dicho agente considera que un ejemplar cualquiera pudiera ser del interés del usuario según sus gustos y preferencias, se producirá la siguiente secuencia de mensajes mediante la cual el Agente Biblioteca le ofrecerá dicho ejemplar y la fecha en que tendría que devolverlo. El usuario contestará con el mensaje de aceptación o rechazo.



Ilustración 31 - Protocolo RecomendaciónEjemplar

LIBRO_RECOMENDADO	
Descripción	Recomendación de un ejemplar que se había solicitado anteriormente o que se considera que puede ser de interés para el usuario.
Performative	QUERY-IF La performative utilizada es QUERY-IF porque este mensaje sirve para <i>preguntar</i> al Usuario si desea aceptar una recomendación.
Emisor	BIBLIOTECA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
ISBN	Código ISBN del ejemplar recomendado.
TITULO	Título del ejemplar recomendado.
AUTOR	Autor del ejemplar recomendado.
FECHA	Fecha en la que debe devolverse el ejemplar en caso de llevárselo.

Tabla 37 - Protocolo RecomendaciónEjemplar (Mensaje 1)

RESPUESTA_RECOMENDACION	
Descripción	Respuesta a la recomendación de un ejemplar.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> a la Biblioteca de si desea o no aceptar la recomendación.
Emisor	ALUMNO/PROFESOR
Receptor	BIBLIOTECA
ARGUMENTOS	
Argumento	Descripción
ID	NIA/PDI del usuario.
ISBN	Código ISBN del ejemplar recomendado.
FECHA	Fecha en que debe devolverse el ejemplar.
RESPUESTA	1 en caso aceptación y 0 en caso de rechazo.

Tabla 38 - Protocolo RecomendaciónEjemplar (Mensaje 2)

Protocolos del Agente Cafetería/Comedor

Notificación de localización de un amigo.

Protocolo: NotificaciónAmigoLocalizado.

Cuando un usuario entra en la zona de Cafetería/Comedor, si un amigo suyo se encuentra también en dicha zona, se le comunica mediante un mensaje.

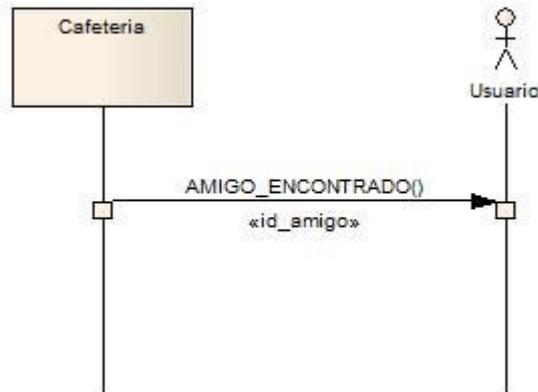


Ilustración 32 - Protocolo NotificaciónAmigoLocalizado

AMIGO_ENCONTRADO	
Descripción	Indica al alumno o profesor que se ha localizado a un amigo suyo.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario de que se ha encontrado a un amigo.
Emisor	CAFETERÍA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
ID_AMIGO	NIA/PDI del amigo encontrado.

Tabla 39 - Protocolo NotificaciónAmigoLocalizado (Mensaje 1)

Notificación de reserva de mesas.

Protocolo: NotificaciónReservaMesa.

Cuando un usuario entra en la zona de Cafetería/Comedor puede solicitar que se le reserve una mesa en una fecha y hora determinadas. En ese caso, la cafetería emitirá el siguiente mensaje indicando los datos de la reserva. En caso de no haber mesas disponibles se devolverá el mensaje con el valor "0" en el número de mesa.

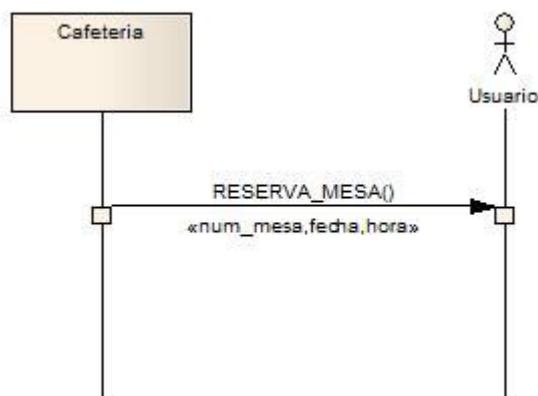


Ilustración 33 - Protocolo NotificaciónReservaMesa

RESERVA_MESA	
Descripción	Notificación de reserva de mesa en la Cafetería/Comedor
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> al Usuario de si le ha sido o no reservada una mesa.
Emisor	ALUMNO/PROFESOR
Receptor	CAFETERÍA
ARGUMENTOS	
Argumento	Descripción
NUM_MESA	Número de mesa. En caso de mesa no disponible este campo tendrá el valor 0.
FECHA	Fecha de la reserva.
HORA	Hora de la reserva.

Tabla 40 - Protocolo NotificaciónReservaMesa (Mensaje 1)

Recordatorio de reserva de mesas.

Protocolo: RecordatorioReservaMesa.

Cuando un usuario entra en la zona de Cafetería/Comedor si tiene alguna mesa reservada, se le enviará un mensaje recordándole el número de mesa, la fecha y hora de la reserva.

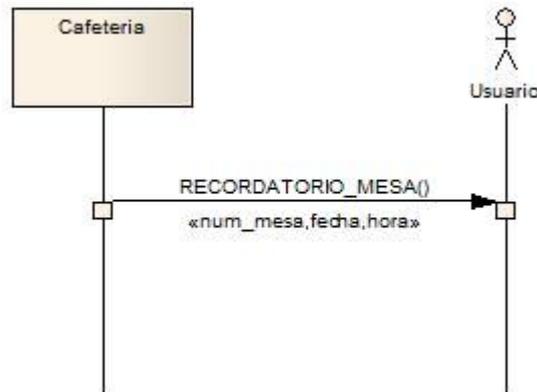


Ilustración 34 - Protocolo RecordatorioReservaMesa

RECORDATORIO_MESA	
Descripción	Recordatorio de reserva de mesa en la Cafetería/Comedor
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre una mesa que tiene reservada.
Emisor	ALUMNO/PROFESOR
Receptor	CAFETERÍA
ARGUMENTOS	
Argumento	Descripción
NUM_MESA	Número de mesa.
FECHA	Fecha de la reserva.
HORA	Hora de la reserva.

Tabla 41 - Protocolo RecordatorioReservaMesa (Mensaje 1)

Recomendación de menús

Protocolo: RecomendaciónMenús.

Cuando un usuario entra en la zona de Cafetería/Comedor a mediodía este agente puede recomendarle el menú de comida que le parezca más adecuado para él según sus gustos y preferencias. Entonces se producirá la siguiente secuencia de mensajes mediante la cual el Agente Cafetería/Comedor le ofrecerá dicho menú y la descripción de lo que contiene y el usuario le contesta aceptando o denegando su recomendación con el mensaje genérico de aceptación/rechazo de consumiciones.



Ilustración 35 - Protocolo RecomendaciónMenús

RECOMENDACION_MENU	
Descripción	Recomendación de un menú.
Performative	QUERY-IF La performative utilizada es QUERY-IF porque este mensaje sirve para <i>preguntar</i> al Usuario si desea aceptar una recomendación.
Emisor	CAFETERÍA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
CODIGO	Código del menú
PRIMERO	Descripción del primer plato.
SEGUNDO	Descripción del segundo plato.
POSTRE	Descripción del postre.
PROPIEDADES	Otras propiedades del menú.

Tabla 42 - Protocolo RecomendaciónMenús (Mensaje 1)

RESPUESTA_RECOMENDACION_CAFETERIA	
Descripción	Respuesta a cualquier recomendación de cafetería, ya sea menú, desayuno o cualquier otra consumición.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> a la Cafetería de si acepta o no la recomendación.
Emisor	ALUMNO/PROFESOR
Receptor	CAFETERÍA
ARGUMENTOS	
Argumento	Descripción
ID	NIA/PDI del usuario.
TIPO	Tipo de recomendación: MENU, DESAYUNO, COMIDA o BEBIDA.
CÓDIGO	Código de consumición.
RESPUESTA	1 en caso aceptación y 0 en caso de rechazo.

Tabla 43 - Protocolo RecomendaciónMenús (Mensaje 2)

Recomendación de desayunos

Protocolo: RecomendaciónDesayunos.

Cuando un usuario entra en la zona de Cafetería/Comedor por la mañana este agente puede recomendarle el desayuno que le parezca más adecuado para él según sus gustos y preferencias. Entonces se producirá la siguiente secuencia de mensajes mediante la cual el Agente Cafetería/Comedor le ofrecerá dicho desayuno y la descripción de lo que contiene y el usuario le contesta aceptando o denegando su recomendación con el mensaje genérico de aceptación/rechazo de consumiciones.

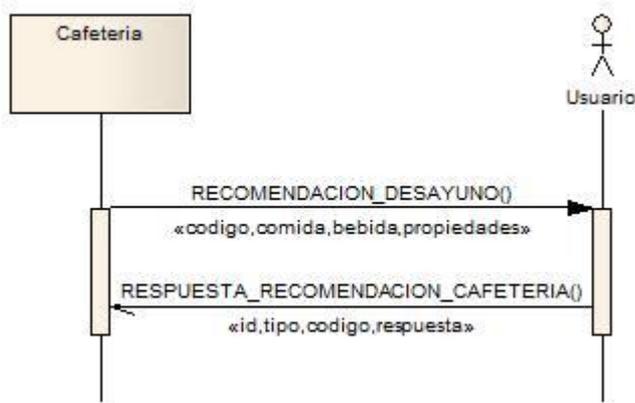


Ilustración 36 - Protocolo RecomendaciónDesayunos

RECOMENDACION_DESAYUNO	
Descripción	Recomendación de un desayuno.
Performative	QUERY-IF La performative utilizada es QUERY-IF porque este mensaje sirve para <i>preguntar</i> al Usuario si desea aceptar una recomendación.
Emisor	CAFETERÍA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
CODIGO	Código del desayuno
COMIDA	Descripción de la comida.
BEBIDA	Descripción de la bebida
PROPIEDADES	Otras propiedades del desayuno.

Tabla 44 - Protocolo RecomendaciónDesayunos (Mensaje 1)

RESPUESTA_RECOMENDACION_CAFETERIA	
Descripción	Respuesta a cualquier recomendación de cafetería, ya sea menú, desayuno o cualquier otra consumición.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> a la Cafetería de si acepta o no la recomendación.
Emisor	ALUMNO/PROFESOR
Receptor	CAFETERÍA
ARGUMENTOS	
Argumento	Descripción
ID	NIA/PDI del usuario.
TIPO	Tipo de recomendación: MENU, DESAYUNO, COMIDA o BEBIDA.
CÓDIGO	Código de consumición.
RESPUESTA	1 en caso aceptación y 0 en caso de rechazo.

Tabla 45 - Protocolo RecomendaciónDesayunos (Mensaje 2)

Recomendación de consumiciones

Protocolo: RecomendaciónConsumiciones.

Cuando un usuario entra en la zona de Cafetería/Comedor por la tarde, o si es por la mañana o mediodía pero no se ha encontrado un desayuno o menú acorde a sus gustos, este agente puede recomendarle la consumición (comida o bebida) que le parezca más adecuada para él según sus gustos y preferencias. Entonces se producirá la siguiente secuencia de mensajes mediante la cual el Agente Cafetería/Comedor le ofrecerá dicha consumición y la descripción de lo que contiene y el usuario le contesta aceptando o denegando su recomendación con el mensaje genérico de aceptación/rechazo de consumiciones.

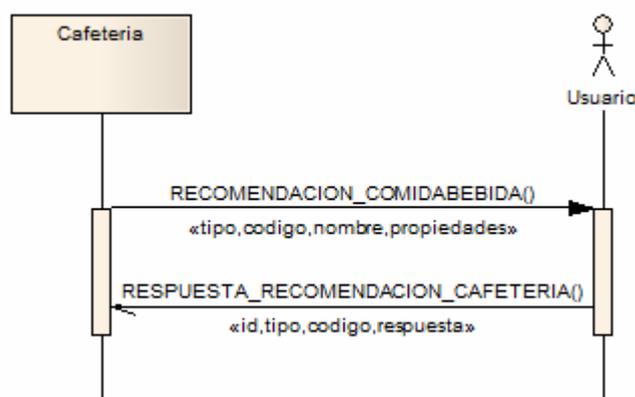


Ilustración 37 - Protocolo RecomendaciónConsumiciones

RECOMENDACION_COMIDABEBIDA	
Descripción	Recomendación de una consumición.
Performative	QUERY-IF La performative utilizada es QUERY-IF porque este mensaje sirve para <i>preguntar</i> al Usuario si desea aceptar una recomendación.
Emisor	CAFETERÍA
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
TIPO	Tipo de consumición: COMIDA o BEBIDA.
CODIGO	Código de la consumición.
NOMBRE	Descripción de la consumición.
PROPIEDADES	Otras propiedades de la consumición.

Tabla 46 - Protocolo RecomendaciónConsumiciones (Mensaje 1)

RESPUESTA_RECOMENDACION_CAFETERIA	
Descripción	Respuesta a cualquier recomendación de cafetería, ya sea menú, desayuno o cualquier otra consumición.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> a la Cafetería de si acepta o no la recomendación.
Emisor	ALUMNO/PROFESOR
Receptor	CAFETERÍA
ARGUMENTOS	
Argumento	Descripción
ID	NIA/PDI del usuario.
TIPO	Tipo de recomendación: MENU, DESAYUNO, COMIDA o BEBIDA.
CÓDIGO	Código de consumición.
RESPUESTA	1 en caso aceptación y 0 en caso de rechazo.

Tabla 47 - Protocolo RecomendaciónConsumiciones (Mensaje 2)

Tablón de anuncios - cambio de turno (solo para alumnos)

Protocolo: TablónCambioTurnos.

Cuando un alumno entra en la zona de Cafetería/Comedor, si este ha realizado una solicitud al Tablón de Anuncios acerca de un cambio de turno, recibirá el siguiente mensaje del Agente Cafetería/Comedor donde se le indica si se ha encontrado un alumno apto para el cambio o si por el contrario la solicitud ha sido anotada.



Ilustración 38 - Protocolo TablónCambioTurnos

INTERCAMBIO_TURNO	
Descripción	Avisa al alumno de si se ha encontrado un alumno para cambiar de turno o si por el contrario la solicitud ha sido anotada.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> al Alumno de si se ha realizado un intercambio o de si este ha sido anotado.
Emisor	CAFETERÍA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
ESTADO	Estado de la solicitud: ENCONTRADO o ANOTADO.
CURSO	Curso académico para el cual se va a realizar el cambio de turno.
TURNO_ACTUAL	Turno actual del alumno solicitante.
TURNO_PEDIDO	Turno pedido por el alumno solicitante.
NIA_ALUMNO	NIA del alumno encontrado para el cambio. Este campo tendrá el valor 0 para el caso de anotar la solicitud porque aun no se ha encontrado cambio posible.

Tabla 48 - Protocolo TablónCambioTurnos (Mensaje 1)

Tablón de anuncios - cambio de libros (solo para alumnos)

Protocolo: TablónCambioLibros.

Cuando un alumno entra en la zona de Cafetería/Comedor, si este ha realizado una solicitud al Tablón de Anuncios acerca de un cambio de libros, recibirá el siguiente mensaje del Agente Cafetería/Comedor donde se le indica si se ha encontrado un alumno apto para el cambio o si por el contrario la solicitud ha sido anotada.

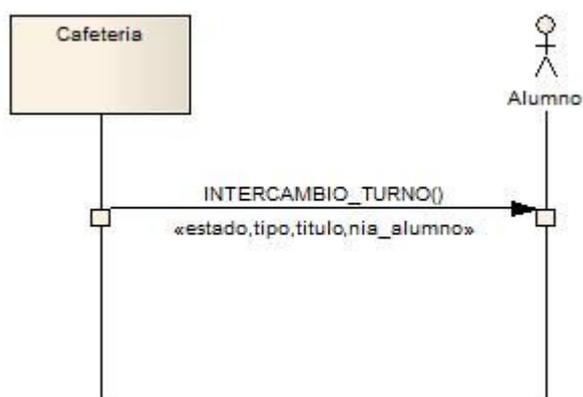


Ilustración 39 - Protocolo TablónCambioLibros

INTERCAMBIO_LIBRO	
Descripción	Avisa al alumno de si se ha encontrado un alumno para cambiar de libro o si por el contrario la solicitud ha sido anotada.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> al Alumno de si se ha realizado un intercambio o de si este ha sido anotado.
Emisor	CAFETERÍA
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
ESTADO	Estado de la solicitud: ENCONTRADO o ANOTADO.
TIPO	Tipo de solicitud: OFERTA o DEMANDA.
TITULO	Título del libro que se desea intercambiar.
NIA_ALUMNO	NIA del alumno encontrado para el cambio. Este campo tendrá el valor 0 para el caso de anotar la solicitud porque aun no se ha encontrado cambio posible.

Tabla 49 - Protocolo TablónCambioLibros (Mensaje 1)

Protocolos del Agente Centro Cultural/Deportivo

Servicio de reserva de entradas de eventos culturales/deportivos

Protocolo: ReservaEntradasEventos.

Cuando un usuario entra en la zona del Centro Cultural/Deportivo puede solicitar la reserva de entradas para un evento cultural o deportivo determinado. Para ello, el Agente Centro Cultural/Deportivo le indicará mediante el siguiente mensaje si ha sido posible realizar la reserva o no.



Ilustración 40 - Protocolo ReservaEntradasEventos

RESERVA_EVENTO	
Descripción	Respuesta sobre si se ha podido reservar entradas.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> al Usuario de si se le ha realizado o no una reserva.
Emisor	CENTRO CULTURAL/DEPORTIVO
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
ESTADO	Estado de la reserva: RESERVADO, PASADO NO_DISPONIBLE o INEXISTENTE.
EVENTO	Código del evento.
TIPO	Tipo de evento: CULTURAL/DEPORTIVO.
CANTIDAD	Número de entradas que se han reservado. En caso de respuesta negativa este valor será 0.

Tabla 50 - Protocolo ReservaEntradasEventos (Mensaje 1)

Servicio de reserva de instalaciones

Protocolo: ReservaInstalaciones.

Cuando un usuario entra en la zona del Centro Cultural/Deportivo puede solicitar la reserva de una instalación en una fecha y hora determinadas. Mediante el siguiente mensaje el Agente Centro Cultural/Deportivo le indicará si la instalación ha sido reservada. Todas las reservas de instalaciones serán de una hora y si se desea más tiempo deberá hacerse una nueva reserva diferente.



Ilustración 41 - Protocolo ReservaInstalaciones

RESERVA_INSTALACION	
Descripción	Respuesta sobre si se ha podido reservar la instalación.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> al Usuario de si se le ha realizado o no una reserva.
Emisor	CENTRO CULTURAL/DEPORTIVO
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
ESTADO	Estado de la reserva: RESERVADA, NO_DISPONIBLE o INEXISTENTE.
INSTALACION	Nombre de la instalación.
FECHA	Fecha de la reserva.
HORA	Hora de la reserva

Tabla 51 - Protocolo ReservaInstalaciones (Mensaje 1)

Recordatorio de entradas reservadas

Protocolo: RecordatorioEntradasReservadas.

Cuando un usuario entra en la zona del Centro Cultural/Deportivo, si tiene reservadas entradas para un evento, recibirá el siguiente mensaje del Agente Centro Cultural/Deportivo donde se le indica que evento es y cuantas entradas tiene reservadas.



Ilustración 42 - Protocolo RecordatorioEntradasReservadas

RECORDATORIO_EVENTO	
Descripción	Recuerda al usuario que tiene un evento reservado.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre un evento reservado.
Emisor	CENTRO CULTURAL/DEPORTIVO
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
EVENTO	Código del evento.
TIPO	Tipo de evento: CULTURAL/DEPORTIVO
CANTIDAD	Número de entradas que se tienen reservadas.

Tabla 52 - Protocolo RecordatorioEntradasReservadas (Mensaje 1)

Recordatorio de instalaciones reservadas

Protocolo: RecordatorioInstalacionesReservadas.

Cuando un usuario entra en la zona del Centro Cultural/Deportivo, si tiene reservada alguna instalación, recibirá el siguiente mensaje del Agente Centro Cultural/Deportivo donde se le indica que instalación es y para que fecha y hora está reservada.



Ilustración 43 - Protocolo RecordatorioInstalacionesReservadas

RECORDATORIO_INSTALACION	
Descripción	Recuerda al usuario la fecha y hora de reserva de una instalación
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Usuario sobre una instalación reservada.
Emisor	CENTRO CULTURAL/DEPORTIVO
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
INSTALACIÓN	Nombre de la instalación para la que se tiene reserva.
FECHA	Fecha de la reserva.
HORA	Hora de la reserva.

Tabla 53 - Protocolo RecordatorioInstalacionesReservadas (Mensaje 1)

Recordatorio de días sin acudir al gimnasio/piscina (solo para alumnos)

Protocolo: RecordatorioDíasAusencia.

Cuando un alumno entra en la zona del Centro Cultural/Deportivo, si lleva más de tres días sin acudir al gimnasio o a la piscina, recibirá el siguiente mensaje del Agente Centro Cultural/Deportivo donde se le indican los días que lleva sin acudir al gimnasio o a la piscina según corresponda.

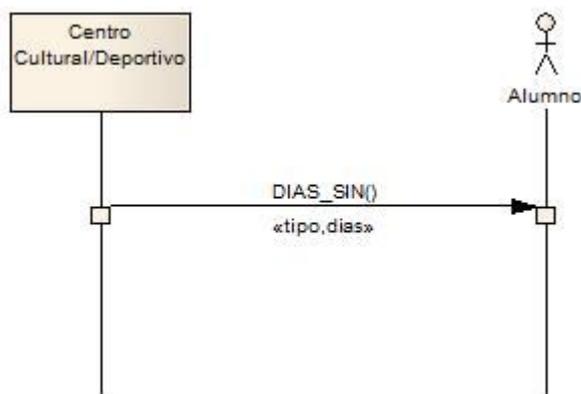


Ilustración 44 - Protocolo RecordatorioDíasAusencia

DIAS_SIN	
Descripción	Indica el número de días que lleva el usuario sin ir al gimnasio o a la piscina.
Performative	INFORM La performative utilizada es INFORM porque este mensaje sirve para <i>informar</i> al Alumno de los días que lleva sin acudir al gimnasio o piscina.
Emisor	CENTRO CULTURAL/DEPORTIVO
Receptor	ALUMNO
ARGUMENTOS	
Argumento	Descripción
TIPO	Lugar al que se refiere el mensaje: GIMNASIO/PISCINA.
DÍAS	Número de días sin acudir.

Tabla 54 - Protocolo RecordatorioDíasAusencia (Mensaje 1)

Recomendación de eventos culturales / Recomendación de eventos deportivos

Protocolo: RecomendaciónEventos.

Cuando un usuario entra en la zona del Centro Cultural/Deportivo este agente puede recomendarle eventos culturales o deportivos que le parezcan adecuados para él según sus gustos y preferencias. Entonces se producirá la siguiente secuencia de mensajes mediante la cual el Agente Centro Cultural/Deportivo le ofrecerá dicho evento y la descripción de este con su fecha, lugar, entradas disponibles, etc. y el usuario le contesta aceptando o denegando su recomendación. En caso de aceptar, deberá indicar cuantas entradas desea.

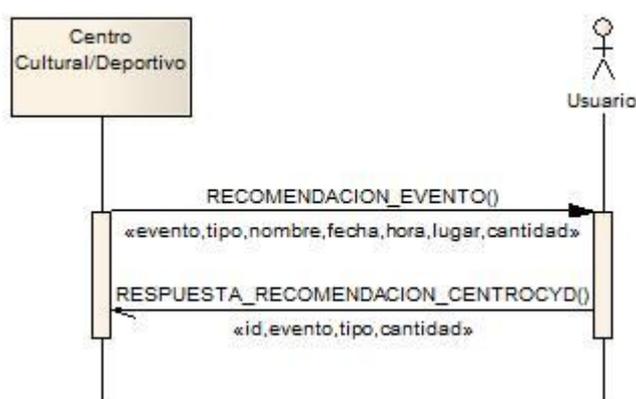


Ilustración 45 - Protocolo RecomendaciónEventos

RECOMENDACION_EVENTO	
Descripción	Recomendación de una actividad cultural o deportiva
Performative	QUERY-IF La performative utilizada es QUERY-IF porque este mensaje sirve para <i>preguntar</i> al Usuario si desea aceptar una recomendación.
Emisor	CENTRO CULTURAL/DEPORTIVO
Receptor	ALUMNO/PROFESOR
ARGUMENTOS	
Argumento	Descripción
EVENTO	Código del evento.
TIPO	Tipo de evento: CULTURAL/DEPORTIVO.
NOMBRE	Nombre del evento.
FECHA	Fecha del evento.
HORA	Hora del evento.
LUGAR	Lugar del evento.
CANTIDAD	Número de entradas disponibles.

Tabla 55 - Protocolo RecomendaciónEventos (Mensaje 1)

RESPUESTA_RECOMENDACION_CENTROCYD	
Descripción	Respuesta a la recomendación.
Performative	INFORM-IF La performative utilizada es INFORM-IF porque este mensaje sirve para <i>informar</i> al CentroCYD de si acepta o no la recomendación.
Emisor	ALUMNO/PROFESOR
Receptor	CENTRO CULTURAL/DEPORTIVO
ARGUMENTOS	
Argumento	Descripción
ID	NIA/PDI del usuario.
EVENTO	Código del evento.
TIPO	Tipo del evento: CULTURAL/DEPORTIVO.
CANTIDAD	Número de entradas deseadas. En caso de que este campo sea 0 la recomendación se tomará como rechazada.

Tabla 56 - Protocolo RecomendaciónEventos (Mensaje 2)

6. Ejecución del escenario de uso con MUSSUS

A continuación se muestra la ejecución del escenario de uso que se describió al comienzo de este documento. En primer lugar podemos ver como al ejecutar el sistema nos encontramos con la ventana maestra, las cuatro zonas y las ventanas de los usuarios; estas últimas superpuestas unas sobre otras.

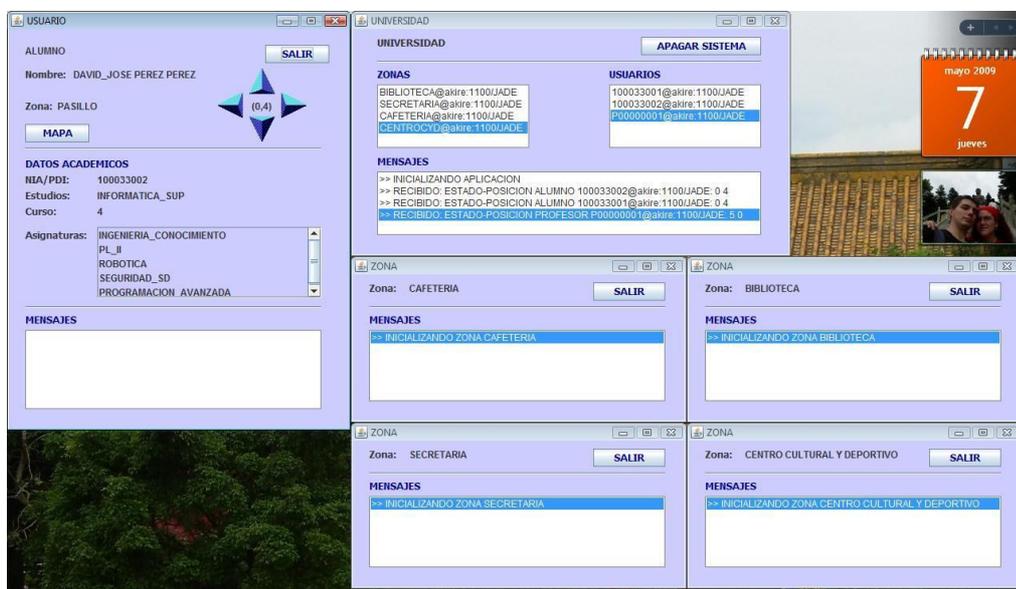


Ilustración 46 - Ejecución de MUSSUS

“David llega a la universidad y conecta el sistema en su teléfono móvil.”



Ilustración 47 - Pantalla de Alumno 1

“Como la entrada está al lado la Secretaría lo primero que recibe es un mensaje indicándole que ha obtenido un 9.0 en la asignatura de “Ingeniería de Conocimiento”; el sistema le ofrece poner una reclamación pero como David está satisfecho con su nota decide cancelar el ofrecimiento. Además la Secretaría le recuerda que tiene pendiente la entrega de la práctica de “Programación Avanzada”.”

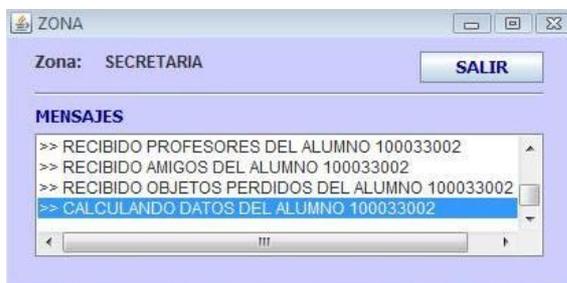


Ilustración 48 - Pantalla Secretaría (Cálculo)



Ilustración 49 - Pantalla Calificación

“A continuación decide irse a desayunar a la Cafetería. Nada más entrar el sistema le ofrece unas patatas bravas que David acepta muy gustosamente, ya que no le entusiasma el dulce.”

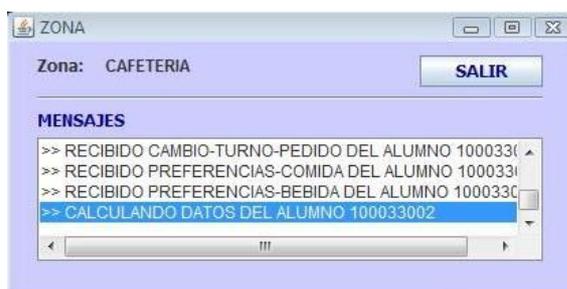


Ilustración 50 - Pantalla Cafetería (Cálculo)



Ilustración 51 - Pantalla Recomendación Cafetería

“Quince minutos después, Eva llega a la universidad y conecta el sistema en su PDA.”



Ilustración 52 - Pantalla Alumno 2

“La primera zona por la que pasa todos los días es la de Secretaría y, al entrar en esta, el sistema le comunica que ha salido la nota del examen de “Redes de Ordenadores II” que estaba esperando y que su calificación es de 5.0. El sistema le ofrece la posibilidad de presentar una reclamación y Eva, al haber obtenido una calificación muy baja, la acepta y la envía con su queja correspondiente. Además el agente de la Secretaría le comunica que han encontrado una cartera roja que podría ser la que Eva había perdido hace unos días.”



Ilustración 53 - Pantalla Calificación (Reclamación)

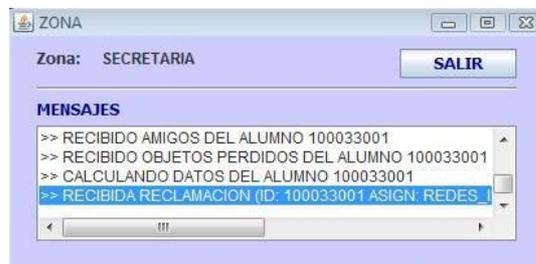


Ilustración 54 - Pantalla Secretaria (Reclamación)

“Antes de ir a clase Eva acude a la Cafetería en busca de su amigo David y al entrar en dicha zona el agente correspondiente le indica automáticamente que este se encuentra ya allí y, a continuación, le recomienda desayunar un donut y un café con leche, aunque en esta ocasión Eva lo rechaza porque tiene prisa.”

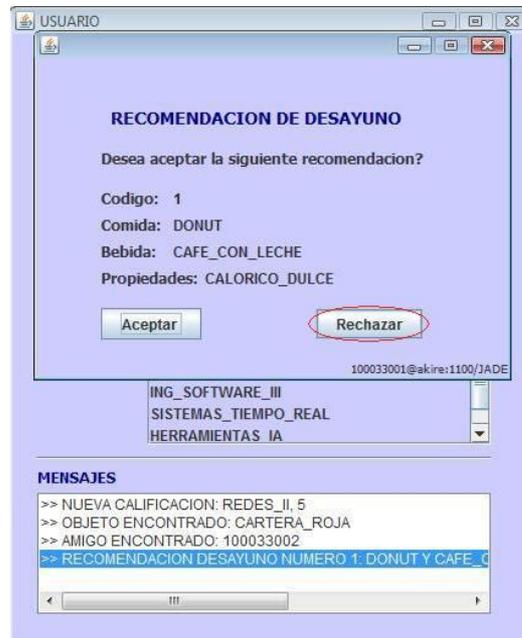


Ilustración 55 - Pantalla Recomendación Desayuno

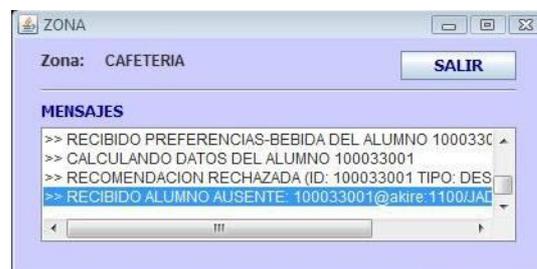


Ilustración 56 - Pantalla Cafetería (Recomendación)

“Después de las clases, va a la Biblioteca a devolver un libro, cosa que le es avisada automáticamente nada más entrar en la Biblioteca. Además le llega la notificación de que ha entrado un nuevo libro de Laura Gallego, su autora favorita, que podría interesarle y Eva lo acepta y se lo lleva.”

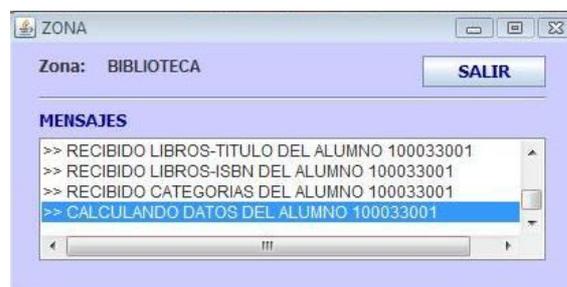


Ilustración 57 - Pantalla Biblioteca (Cálculo)



Ilustración 58 - Pantalla Recomendación Libro

“Al finalizar el día Eva pasa por el Centro Cultural y Deportivo para recoger unas entradas para un partido de fútbol. Nada más entrar, el sistema le recuerda que lleva 12 días sin pasar por el gimnasio. A cambio de la regañina le recomienda un espectáculo de teatro para el viernes por la noche, que por supuesto acepta para ella y otras 3 personas, volviendo a casa con planes para el fin de semana.”



Ilustración 59 - Pantalla Recomendación Evento



Ilustración 60 - Pantalla CentroCyD (Recomendación)

Al salir del perímetro de la universidad Eva se desconecta del sistema.



Ilustración 61 - Pantalla Usuario (Salir)

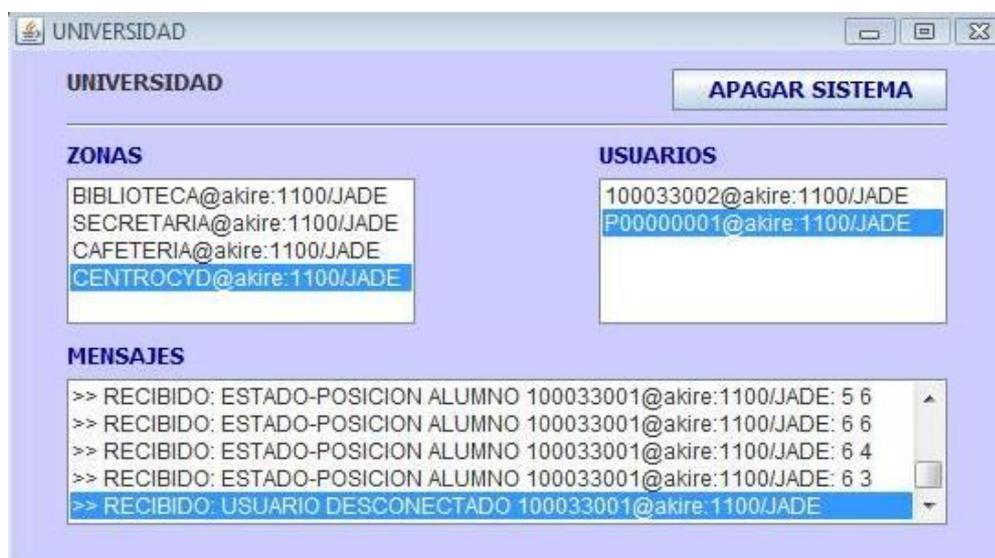


Ilustración 62 - Pantalla Universidad (Usuario Desconectado)

“Esa misma tarde, Francisco termina de impartir sus clases y decide pasar por Secretaría como cada tarde para comprobar su casillero. Como es nuevo en la universidad a menudo se desorienta, por lo que consulta el mapa para ver donde se encuentra.”



Ilustración 63 - Pantalla Profesor



Ilustración 64 - Pantalla Mapa

“Cuando llega a la Secretaría se le comunica que tiene 5 entregas en su casillero, además de la reclamación que Eva le había presentado esa misma mañana. Además se le indica que un profesor al que estaba buscando se encuentra en su despacho en este momento.”

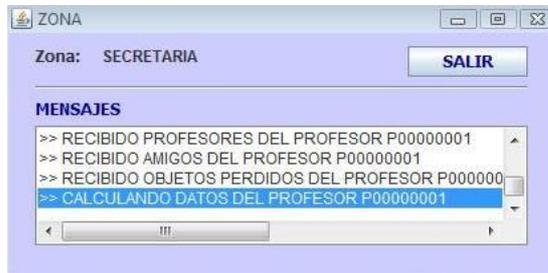


Ilustración 65 - Pantalla Secretaría (Cálculo Profesor)



Ilustración 66 - Pantalla Profesor Secretaría

Finalmente, el sistema puede cerrarse desde el botón correspondiente del Agente Universidad. De este modo, dicho agente avisará a las zonas y usuarios aún conectados de que deben volcar sus datos y desconectarse.

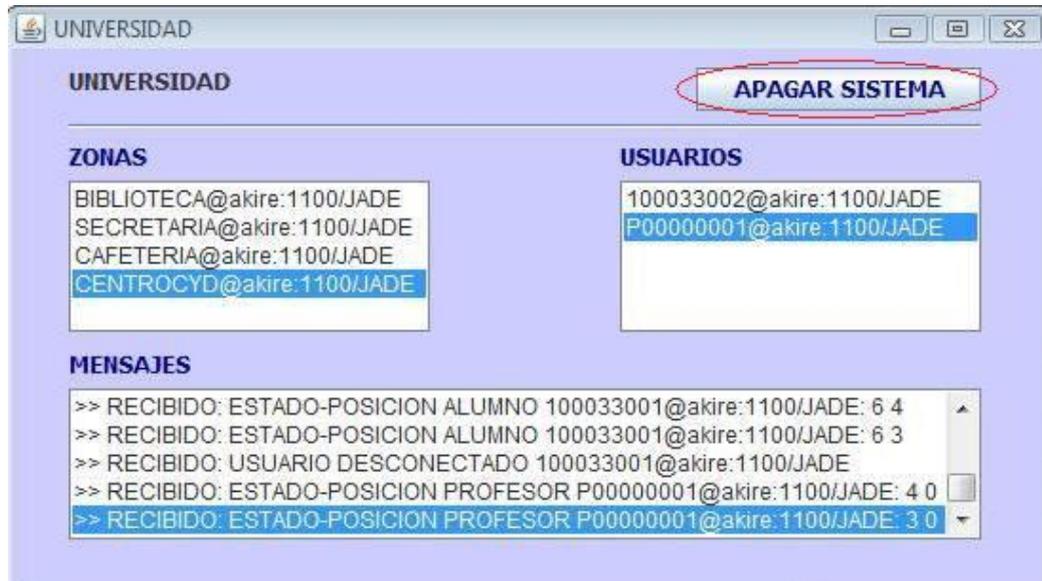


Ilustración 67 - Pantalla Apagar Sistema

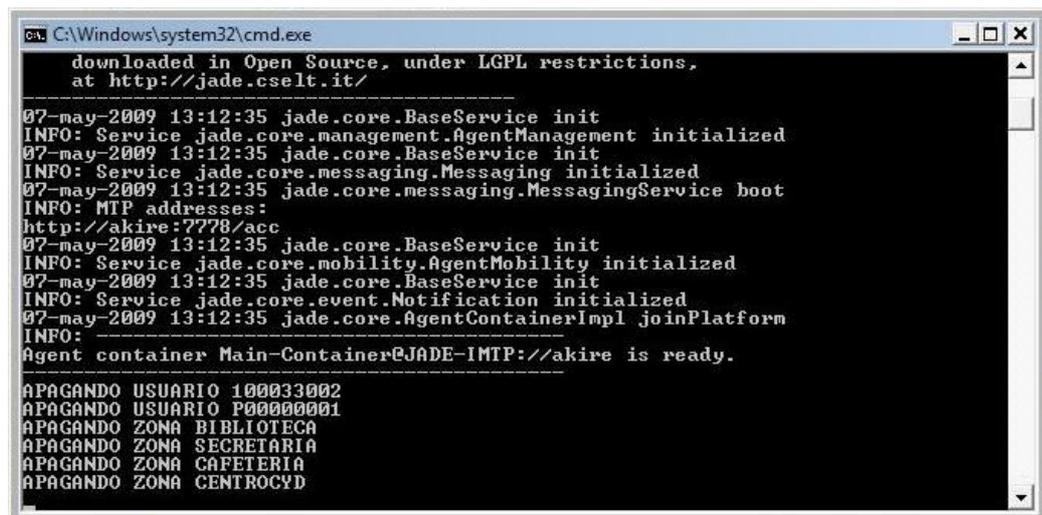


Ilustración 68 - Apagado de agentes

7. Conclusiones

Para finalizar, se expondrán las conclusiones obtenidas gracias al análisis, diseño e implementación del sistema MUSSUS.

En primer lugar se comentarán los principales problemas encontrados, tanto a la hora del análisis y diseño como en la implementación del sistema, explicando además las medidas tomadas para remediarlos.

Una vez solventados los problemas, se explicará como este sistema puede ser adaptado fácilmente a muchos otros dominios, que no tienen porque tener relación con el mundo de la educación.

Finalmente, se comentarán algunas ideas sobre posibles reformas o ampliaciones que se podrían llevar a cabo en torno a este proyecto, en caso de contar con más medios.

Problemas encontrados.

Además de los típicos problemas propios de la implementación de cualquier aplicación, se han encontrado algunos problemas en el desarrollo de este Sistema Multiagente que se verán a continuación.

Localización de los usuarios

Uno de los primeros problemas que se encontraron al diseñar MUSSUS fue el como llevar a cabo la localización de los usuarios en las zonas del sistema.

En un entorno real, los usuarios serían fácilmente detectables mediante tecnologías inalámbricas como el Bluetooth, pero en el entorno simulado de MUSSUS el mapa es únicamente una cuadrícula y era necesario simular, de algún modo, el movimiento y la entrada y salida de las zonas de los usuarios.

Para solventar este problema se creó el Agente Universidad, que hace las funciones del maestro del sistema. De este modo, el Agente Universidad identifica en que zona se encuentra cada usuario según su posición y, en caso de estar dentro de una zona activa, pondrá en contacto al usuario con dicha zona. En un entorno real todo esto no sería necesario, ya que la propia zona podría detectar y ponerse en contacto con el usuario en cuanto este entrase dentro de su radio de actuación.

Acceso reiterado a los archivos XML

Otro problema, que afectaba al rendimiento del sistema, era la necesidad de acceder y recorrer de manera muy repetida los archivos XML que contienen los datos del sistema. Y más aun, la necesidad de tener que escribir de manera continua en estos archivos mientras el sistema se estaba ejecutando.

Por ello, se tomó la decisión de crear varios vectores en los cuales poder volcar la información de los XML al iniciar el sistema. El resto de operaciones de lectura, escritura, modificación y borrado se harán sobre dichos vectores. Solo cuando el sistema es apagado, la información de los vectores vuelve a ser volcada a los archivos XML permanentes; por ello será muy importante apagar el sistema correctamente.

Diseño de los mensajes

El diseño de los mensajes del sistema es una de las fases fundamentales en el desarrollo de un Sistema Multiagente y, por tanto, también una de las más complejas. Por ello, es muy aconsejable realizar un buen análisis y redactar una ontología del sistema lo más precisa posible.

En este caso en concreto, pudo comprobarse como a pesar de realizar una ontología lo más detallada posible, fue necesario modificar posteriormente el diseño de algunos mensajes que habían quedado incoherentes, incompletos o que debían ser modificados debido a nuevas prestaciones del sistema que no habían sido contempladas en la primera fase de análisis. Un ejemplo de esto es que tuvo que añadirse un campo "tipo" en algunos mensajes que podían ser emitidos tanto por alumnos como profesores para poder distinguir el tipo de usuario que estaba emitiendo el mensaje.

En relación con la estructura de los mensajes también se ha encontrado otro problema que, si bien no es grave, si que apareció en repetidas ocasiones y provocó gran pérdida de tiempo hasta que fue detectado por primera vez. Este problema guarda relación con los parámetros de los mensajes enviados.

Como los mensajes pueden tener un número ilimitado de parámetros es necesario tener una constante que indique cuantos parámetros tiene cada mensaje. El problema viene cuando un determinado parámetro contiene un espacio en blanco, ya que el sistema lo reconoce como otro parámetro distinto y no se corresponde por tanto con el valor de la constante. Esto puede darse, por ejemplo, con los nombres compuestos, los títulos de libros, los nombres de asignaturas, los textos de las reclamaciones, etc.

Para solventar este problema fue necesario convertir todos los espacios en blanco en el carácter "_" antes de enviar un mensaje. Como consecuencia de esto, también en los XML los espacios en blanco deben ser siempre sustituidos por "_".

Apagado de los agentes

A la hora de desconectar a un agente del sistema se encontró un problema fácil de solventar pero desconocido en un principio. Este problema consistía en que, al desconectar un agente del sistema se llamaba al método *doDelete()*, propio de los agentes de JADE y que se utiliza para eliminar dichos agentes del sistema. Lo que en un principio se desconocía es que, a pesar de eliminar el agente y cerrar sus ventanas asociadas, este permanecía en el registro, por lo que el Agente Universidad volvía a detectarlo en el siguiente ciclo y volvía a añadirlo a su vector de agentes/zonas, por lo que en realidad continuaba ejecutándose aunque no fuera visible.

Por otro lado, también se encontró el problema de que al volcar la información de nuevo al XML antes de que el agente se cerrase, a veces se realizaba correctamente y a veces no. Y esto era debido a que, en ocasiones, se tardaba más en volcar dichos datos que en enviar el mensaje de apagar y ejecutar *doDelete()*.

Ambos problemas se solventaron del mismo modo, sobrescribiendo el método *takeDown()*. Este método se emplea para incluir en él todas las acciones que se quieran realizar antes de que un agente se apague y, precisamente, es llamado por el método *doDelete()*. En este caso, en el método *takeDown()* se llamará a la función encargada de actualizar los datos del XML y, posteriormente, se llamará a otra función que ha sido implementada para eliminar definitivamente a los agentes del registro. Al ejecutarse dentro del método *takeDown()*, nos aseguramos de que ambas funciones se ejecutan perfecta y completamente.

Implementación de la Interfaz

En la fase de implementación de la interfaz gráfica se han encontrado varios problemas, todos ellos propios de la utilización de las librerías Swing de JAVA y del IDE Netbeans 6.5.

Los principales problemas están relacionados con el uso de imágenes, como por ejemplo la imagen de fondo y el icono de posición de la ventana del mapa o la colocación de imágenes en los botones de movimiento.

Otro problema fue la utilización de una *JList*, que finalmente tuvo que ser sustituida por una *List*, para mostrar el buffer con todos los mensajes recibidos por un agente. Principalmente, se encontraron problemas a la hora de actualizar la lista y, posteriormente, posicionarla en el último mensaje recibido y que este apareciese destacado.

Todos los problemas relacionados con la interfaz gráfica han sido solucionados gracias a la perseverancia y a la búsqueda de información en la red y en manuales de programación especializados.

Adaptación a otros dominios.

Como ya hemos visto, MUSSUS es un sistema diseñado para proveer a sus usuarios de servicios, dentro de un entorno universitario. Pero, la versatilidad propia de los Sistemas Multiagente, y más concretamente la de este que nos atañe, hace que esta misma idea de sistema pueda ser aplicada a muchos otros dominios.

En otras palabras, MUSSUS ha sido diseñado de manera que, aunque hoy se utilizaría para un entorno universitario, mañana podría ser adaptado muy fácilmente a cualquier otro entorno que no tiene porque tener ninguna relación con la enseñanza o la educación.

Algunos ejemplos de dominios en los que podría utilizarse este sistema son los que se mencionan a continuación:

- ***Hospital:*** un entorno en el que podría emplearse MUSSUS y que sería de gran utilidad es el hospitalario. Así, cada zona del hospital podría ser identificada por un agente distinto (cardiología, cirugía, maternidad, etc.) y los distintos usuarios del sistema podrían dividirse en pacientes y personal sanitario; incluso podría afinarse más y dividir el personal sanitario en médicos, enfermeras, personal de apoyo, etc. ya que no todos tendrían las mismas funciones. Los servicios proporcionados dependerían de cada una de las zonas, pudiendo por ejemplo recordar citas médicas, indicar donde se encuentra un paciente, recordar operaciones, números de consulta, etc. Por supuesto, también podríamos contar con las funciones del Agente Cafetería sin realizar grandes modificaciones en sus funciones salvo las del tablón de anuncios.
- ***Centro Comercial:*** otro entorno al que MUSSUS podría adaptarse fácilmente es al de un centro comercial. De este modo, cada una de las tiendas tendrían su propio agente y los usuarios serían principalmente los clientes y los vendedores. Las funciones podrían ser, por ejemplo, la recomendación de productos, aviso de ofertas, recordatorio de pedidos, registro de visitas, compras, ventas, etc. Nuevamente, podría utilizarse sin grandes modificaciones el Agente Cafetería para cada uno de los establecimientos dedicados a la restauración.

- Hotel: un hotel también podría ser un buen entorno para utilizar MUSSUS, habiendo distintos agentes en cada una de las zonas del hotel (cafetería, restaurante, recepción, zona de piscina, salas de ocio, etc.) Incluso podría tenerse un agente en cada una de las habitaciones para satisfacer más personalmente las necesidades de los clientes, como la llamada al servicio de habitaciones o la petición de limpieza de cuarto, cambio de almohadas, etc. Nuevamente, los agentes usuarios serían principalmente los clientes y el personal del hotel y las funciones del sistema podrían ir desde la reserva de instalaciones o la recomendación de actividades y viajes, hasta el recordatorio de pagos, o la distribución automática de habitaciones, etc.
- Otras: existen otros dominios en los que podría emplearse de manera sencilla el sistema MUSSUS como, por ejemplo, una gran empresa o fábrica en la cual existan varias zonas de operación y en las que los trabajadores puedan clasificarse según su categoría, formando así los distintos tipos de usuarios, con sus distintas funciones. Por otro lado, también podría utilizarse en museos, exposiciones o grandes eventos como podría ser una feria o una exposición mundial. De esta manera, cada sala o zona tendría su propio agente personalizado que podría informar y proveer de sus servicios a los visitantes, que serían los principales usuarios del sistema junto con los trabajadores.

Se han comentado solo algunos ejemplos de aplicación de MUSSUS a otros dominios, pero pueden existir muchos más. Concretamente, este sistema podría ser adaptado a cualquier entorno que precise de proveer a sus usuarios de servicios, información o recomendaciones que puedan realizarse de manera automática y que, preferiblemente, sea un entorno que conste de varias zonas y distintos tipos de usuarios, aunque esto último no es estrictamente necesario.

Trabajo futuro.

El objetivo de este proyecto era la creación de un Sistema Multiagente, centrándonos principalmente en la parte del diseño y definición de la ontología, que es realmente lo más relacionado con la teoría de agentes. Es por ello que el sistema se ha desarrollado de manera simulada y que, aunque se ha intentado realizar la interfaz lo más funcional y vistosa posible, su diseño no ha sido la principal prioridad del proyecto. Además la interfaz en un entorno real debería estar adaptada a los dispositivos, por lo que sería una pérdida de tiempo realizar una interfaz gráfica muy elaborada que luego no podría ser utilizada en un entorno real.

Debido a esto, la mayor parte del trabajo futuro tiene relación con el entorno real y la interfaz gráfica del sistema, aunque existen otras posibles ampliaciones, que no requieren modificar gran volumen de código, ya que el sistema está diseñado para eso. Esas ampliaciones consisten en la creación de nuevos usuarios y zonas para el sistema o la implementación de nuevas funciones que se deseen añadir tanto para los nuevos agentes como para los que ya existían.

Siguiendo con la interfaz gráfica, podrían realizarse varias mejoras o ampliaciones como podría ser la conversión de los “_” nuevamente por espacios en blanco que, como ya se comentó anteriormente, tuvieron que ser introducidos para evitar problemas en los mensajes. Esta conversión solamente se realizaría para conseguir una interfaz más vistosa, sin tener ninguna otra utilidad. Lo que sí podría ser útil es añadir un botón a la interfaz que permita al usuario o a la zona volcar sus datos a los XML en cualquier momento y no solo al salir de la aplicación.

En relación con la interfaz también existe otra mejora que consistiría en crear una pantalla de introducción de datos, que permita a los usuarios añadir información sobre sus gustos, deseos e inquietudes, etc. automáticamente desde la aplicación, no teniendo que manipular directamente los archivos XML, como se hace ahora.

Finalmente, el principal trabajo futuro que podría realizarse en relación con el sistema MUSSUS es su implantación en un entorno real. Esta implantación podría realizarse como un nuevo proyecto educativo, que podría considerarse una continuación de este; pero también podría realizarse como sistema real en la universidad para mejorar la experiencia de los nuevos estudiantes y trabajadores. En cualquiera de los casos, dicha implantación requeriría de muchos medios, tanto económicos como de recursos físicos, personales y, por supuesto, temporales.

Dentro de la implantación en un entorno real sería necesario realizar varias acciones. Una de ellas consistiría en eliminar el agente maestro, ya que ya no sería necesario para la localización de los agentes. Aunque también podría conservarse como maestro, cuya única función sería guardar la relación de las zonas y agentes conectados. En cualquier caso, la localización de los agentes pasaría a ser función de cada una de las zonas.

Una parte muy importante de la implantación real sería la utilización de bases de datos reales. De este modo, los usuarios continuarían almacenando sus datos en archivos XML porque, como se explicó al comienzo de este documento, al tratarse de un formato universal puede ser comprendido por cualquier tipo de terminal. Sin embargo, para el caso de los agentes zonas sería necesario adecuarse a las bases de datos (BBDD) ya existentes, como por ejemplo la de la biblioteca o la de secretaría de alumnos.

De este modo y llegados a este punto caben varias opciones: la primera de ellas consistiría en la creación de una aplicación que convierta los datos de la BBDD actual en el formato XML y, de este modo, seguir ejecutando la aplicación como se hace actualmente. La otra opción sería crear una nueva BBDD para cada una de las zonas, que obtenga sus datos maestros de las BBDD ya existentes en la actualidad y que contenga también los datos necesarios para ejecutar MUSSUS. En este segundo caso, los datos podrían ser después convertidos al XML o, directamente, se podrían realizar las consultas y operaciones contra estas nuevas BBDD, para lo cual sería necesario retocar el código actual e introducir las correspondientes sentencias o consultas.

Finalmente, si el sistema fuera implantando en un entorno real, sería necesario adecuar la interfaz de usuario a los distintos dispositivos con los que se pueda ejecutar el sistema, ya que no es lo mismo ejecutar la aplicación en un teléfono, que en una PDA o que en un ordenador portátil, etc.

Conclusión final.

Para terminar y como conclusión final solo queda decir que, aunque no se trate de una tecnología muy utilizada actualmente, los Sistemas Multiagente son una muy buena opción para resolver determinados problemas que precisen de división de tareas, cooperación y, sobre todo, de un cierto grado de complejidad. Además se trata de una tecnología cuya aplicación es relativamente sencilla, comparada con otras técnicas, ya que los agentes se pueden asociar fácilmente con el concepto de objeto y, sobre todo y más importante, se pueden relacionar fácilmente con entidades del mundo real.

8. Bibliografía

- [1] Mark Weiser's Home Page, 1996. <http://www.ubiq.com/weiser/>
- [2] Mark Weiser. *The world is not a desktop*. Interactions, pp.7-8, 1994.
- [3] Maes, Pattie. *Artificial Life Meets Entertainment: Life like Autonomous Agents*. Communications of the ACM, 38, 11, 108-114, 1995.
- [4] N.R. Jennings y M. Wooldridge. *Agent-Oriented Software Engineering*. Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, 2000.
- [5] JADE (Java Agent Development Framework), Telecom Italia, 2008. <http://jade.tilab.com/>
- [6] M. Wooldridge, N.R. Jennings y D. Kinny. *The Gaia Methodology for Agent-Oriented Analysis and Design*. Journal of Autonomous Agents and Multi-Agent Systems, 3, 285-312, 2000.
- [7] Xavier Alamán, Pablo Haya, Germán Montoro. *ODISEA: Hacia un entorno inteligente basado en un interfaz en lenguaje natural*. Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, 2000.
- [8] Montserrat Sendín, Toni Granollers, Jesús Lorés, Carles Aguiló, Alexandra Balaguer. *Un modelo interactivo ubicuo aplicado al patrimonio natural y cultural del área del Montsec*. Departamento de Informática e Ingeniería Industrial, Universidad de Lleida, 2002.
- [9] Diego López de Ipiña. *MOBILE EYE: Servicios Móviles Sensibles al Contexto*. Facultad de Ingeniería, Universidad de Deusto, 2004.
- [10] Cristobal Raya, Jose Antonio Soria, Cecilio Angulo. *Sistema de control cooperativo multi-agente en el entorno de una vivienda asistida*. Dep. Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, 2004.
- [11] Pedro Santana. *Computación ubicua para el cuidado de pacientes en estado crítico en el hogar*. Facultad de Ingeniería, Universidad Autónoma de Baja California, 2004.

- [12] Loreto Susperregi, Iñaki Maurtua, Carlos Tubío, Miguel Angel Pérez, Iñigo Segovia, Basilio Sierra. *AMICO - Una arquitectura multiagente para un Laboratorio de Inteligencia Ambiental en Fabricación*. Fundación TEKNIKER, Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad del País Vasco, 2004.

- [13] Angela Carrillo Ramos, Jérôme Gensel, Marlène Villanova-Oliver, Hervé Martin. *PUMAS: Un Framework que Adapta la Información en Ambientes Ubicuos*. LSR-IMAG Laboratory, SIGMA Team, 2005.

- [14] Elaine McGovern, Bernard Joseph Roche, Eleni E. Mangina, Rem W. Collier. *IUMELA: A Lightweight Multi-Agent Systems based Mobile Learning Assistant Using the ABITS Messaging Service. Volume 4611, pgs 1056-1065, Springer-Verlag*. School of Computer Science & Informatics University College Dublin, 2007.

9. Índice de tablas

Tabla 1 - Comparativa de sistemas	22
Tabla 2 - Rol Localizador.....	33
Tabla 3 - Rol Secretaria.....	34
Tabla 4 - Rol Biblioteca	35
Tabla 5 - Rol Cafetería.....	36
Tabla 6 - Rol Comedor.....	37
Tabla 7 - Rol Centro Cultural.....	38
Tabla 8 - Rol Centro Deportivo	39
Tabla 9 - Rol Alumno	40
Tabla 10 - Rol Profesor	41
Tabla 11 - Protocolo MovimientoUsuarios (Mensaje 1).....	47
Tabla 12 - Protocolo MovimientoUsuarios (Mensaje 2).....	48
Tabla 13 - Protocolo MovimientoUsuarios (Mensaje 3).....	48
Tabla 14 - Protocolo MovimientoUsuarios (Mensaje 4).....	48
Tabla 15 - Protocolo EnvíoInformación (Mensaje 1)	50
Tabla 16 - Protocolo EnvíoInformación (Mensaje 2).....	50
Tabla 17 - Protocolo EnvíoInformación (Mensaje 3).....	51
Tabla 18 - Protocolo DesconexiónUsuarios (Mensaje 1)	52
Tabla 19 - Protocolo DesconexiónUsuarios (Mensaje 2)	52
Tabla 20 - Protocolo DesconexiónZonas (Mensaje 1).....	53
Tabla 21 - Protocolo DesconexiónSistema (Mensaje 1).....	54
Tabla 22 - Protocolo NotificaciónCasillero (Mensaje 1)	55
Tabla 23 - Protocolo NotificaciónReclamaciones (Mensaje 1).....	56
Tabla 24 - Protocolo NotificaciónNotas (Mensaje 1).....	57
Tabla 25 - Protocolo NotificaciónNotas (Mensaje 2).....	58
Tabla 26 - Protocolo RecordatorioPlazosEntrega (Mensaje 1)	59
Tabla 27 - Protocolo RecordatorioExámen (Mensaje 1).....	60
Tabla 28 - Protocolo NotificaciónProfesorLocalizado (Mensaje 1).....	61
Tabla 29 - Protocolo NotificaciónAlumnoLocalizado (Mensaje 1)	62
Tabla 30 - Protocolo NotificaciónObjetoPerdido (Mensaje 1).....	63
Tabla 31 - Protocolo RecordatorioEntregaEjemplar (Mensaje 1).....	64
Tabla 32 - Protocolo RecordatorioDíasSanción (Mensaje 1).....	65
Tabla 33 - Protocolo ReservaAulas (Mensaje 1).....	66
Tabla 34 - Protocolo RecordatorioAulaReservada (Mensaje 1).....	67
Tabla 35 - Protocolo RecordatorioLibroReservado (Mensaje 1)	68
Tabla 36 - Protocolo NotificaciónPréstamo (Mensaje 1).....	69
Tabla 37 - Protocolo RecomendaciónEjemplar (Mensaje 1)	70
Tabla 38 - Protocolo RecomendaciónEjemplar (Mensaje 2)	71
Tabla 39 - Protocolo NotificaciónAmigoLocalizado (Mensaje 1)	72
Tabla 40 - Protocolo NotificaciónReservaMesa (Mensaje 1)	73
Tabla 41 - Protocolo RecordatorioReservaMesa (Mensaje 1)	74
Tabla 42 - Protocolo RecomendaciónMenús (Mensaje 1)	75
Tabla 43 - Protocolo RecomendaciónMenús (Mensaje 2).....	76
Tabla 44 - Protocolo RecomendaciónDesayunos (Mensaje 1).....	77
Tabla 45 - Protocolo RecomendaciónDesayunos (Mensaje 2).....	77
Tabla 46 - Protocolo RecomendaciónConsumiciones (Mensaje 1).....	78
Tabla 47 - Protocolo RecomendaciónConsumiciones (Mensaje 2).....	79
Tabla 48 - Protocolo TablónCambioTurnos (Mensaje 1)	80

Tabla 49 - Protocolo TablónCambioLibros (Mensaje 1)	81
Tabla 50 - Protocolo ReservaEntradasEventos (Mensaje 1).....	82
Tabla 51 - Protocolo ReservaInstalaciones (Mensaje 1).....	83
Tabla 52 - Protocolo RecordatorioEntradasReservadas (Mensaje 1)	84
Tabla 53 - Protocolo RecordatorioInstalacionesReservadas (Mensaje 1).....	85
Tabla 54 - Protocolo RecordatorioDíasAusencia (Mensaje 1).....	86
Tabla 55 - Protocolo RecomendaciónEventos (Mensaje 1).....	87
Tabla 56 - Protocolo RecomendaciónEventos (Mensaje 2).....	88
Tabla 57 - Distribución de la carga de trabajo	113

10. Índice de ilustraciones

Ilustración 1 - Funcionamiento de MobileEye	13
Ilustración 2 - Ejemplo de habitación en THC	15
Ilustración 3 - Técnico de AMICO	17
Ilustración 4 - Organización de PUMAS	19
Ilustración 5 - Dispositivo con IUMELA instalado	20
Ilustración 6 - Correspondencia entre roles y agentes	42
Ilustración 7 - Conceptos generales	43
Ilustración 8 - Conceptos del agente secretaría	43
Ilustración 9 - Conceptos del agente biblioteca	44
Ilustración 10 - Conceptos del agente cafetería/comedor	44
Ilustración 11 - Conceptos del agente centro cultural/deportivo	45
Ilustración 12 - Protocolo MovimientoUsuarios	47
Ilustración 13 - Protocolo EnvíoInformación	49
Ilustración 14 - Protocolo DesconexiónUsuarios	51
Ilustración 15 - Protocolo DesconexiónZonas	53
Ilustración 16 - Protocolo DesconexiónSistema	54
Ilustración 17 - Protocolo NotificaciónCasillero	55
Ilustración 18 - Protocolo NotificaciónReclamaciones	56
Ilustración 19 - Protocolo NotificaciónNotas	57
Ilustración 20 - Protocolo RecordatorioPlazosEntrega	58
Ilustración 21 - Protocolo RecordatorioExámen	59
Ilustración 22 - Protocolo NotificaciónProfesorLocalizado	60
Ilustración 23 - Protocolo NotificaciónAlumnoLocalizado	61
Ilustración 24 - Protocolo NotificaciónObjetoPerdido	62
Ilustración 25 - Protocolo RecordatorioEntregaEjemplar	64
Ilustración 26 - Protocolo RecordatorioDíasSanción	65
Ilustración 27 - Protocolo ReservaAulas	66
Ilustración 28 - Protocolo RecordatorioAulaReservada	67
Ilustración 29 - Protocolo RecordatorioLibroReservado	68
Ilustración 30 - Protocolo NotificaciónPréstamo	69
Ilustración 31 - Protocolo RecomendaciónEjemplar	70
Ilustración 32 - Protocolo NotificaciónAmigoLocalizado	72
Ilustración 33 - Protocolo NotificaciónReservaMesa	73
Ilustración 34 - Protocolo RecordatorioReservaMesa	74
Ilustración 35 - Protocolo RecomendaciónMenús	75
Ilustración 36 - Protocolo RecomendaciónDesayunos	76
Ilustración 37 - Protocolo RecomendaciónConsumiciones	78
Ilustración 38 - Protocolo TablónCambioTurnos	79
Ilustración 39 - Protocolo TablónCambioLibros	80
Ilustración 40 - Protocolo ReservaEntradasEventos	82
Ilustración 41 - Protocolo ReservaInstalaciones	83
Ilustración 42 - Protocolo RecordatorioEntradasReservadas	84
Ilustración 43 - Protocolo RecordatorioInstalacionesReservadas	85
Ilustración 44 - Protocolo RecordatorioDíasAusencia	86
Ilustración 45 - Protocolo RecomendaciónEventos	87
Ilustración 46 - Ejecución de MUSSUS	89
Ilustración 47 - Pantalla de Alumno 1	89
Ilustración 48 - Pantalla Secretaría (Cálculo)	90

Ilustración 49 - Pantalla Calificación.....	90
Ilustración 50 - Pantalla Cafetería (Cálculo).....	90
Ilustración 51 - Pantalla Recomendación Cafetería.....	91
Ilustración 52 - Pantalla Alumno 2.....	91
Ilustración 53 - Pantalla Calificación (Reclamación).....	92
Ilustración 54 - Pantalla Secretaria (Reclamación).....	92
Ilustración 55 - Pantalla Recomendación Desayuno.....	93
Ilustración 56 - Pantalla Cafetería (Recomendación).....	93
Ilustración 57 - Pantalla Biblioteca (Cálculo).....	93
Ilustración 58 - Pantalla Recomendación Libro.....	94
Ilustración 59 - Pantalla Recomendación Evento.....	94
Ilustración 60 - Pantalla CentroCyD (Recomendación).....	95
Ilustración 61 - Pantalla Usuario (Salir).....	95
Ilustración 62 - Pantalla Universidad (Usuario Desconectado).....	95
Ilustración 63 - Pantalla Profesor.....	96
Ilustración 64 - Pantalla Mapa.....	96
Ilustración 65 - Pantalla Secretaría (Cálculo Profesor).....	97
Ilustración 66 - Pantalla Profesor Secretaría.....	97
Ilustración 67 - Pantalla Apagar Sistema.....	98
Ilustración 68 - Apagado de agentes.....	98
Ilustración 69 - Diagrama src.....	110
Ilustración 70 - Diagrama gui.....	110
Ilustración 71 - Diagrama almacenamiento.....	111
Ilustración 72 - Diagrama comunicaciones.....	111
Ilustración 73 - Diagrama agentes.....	112

Anexo A – Diagramas de Clases

El sistema consta de cuatro paquetes que separan las distintas clases según sus funcionalidades, que pueden estar dedicadas a la interfaz de usuario, al almacenamiento, a las comunicaciones o a la definición de agentes.

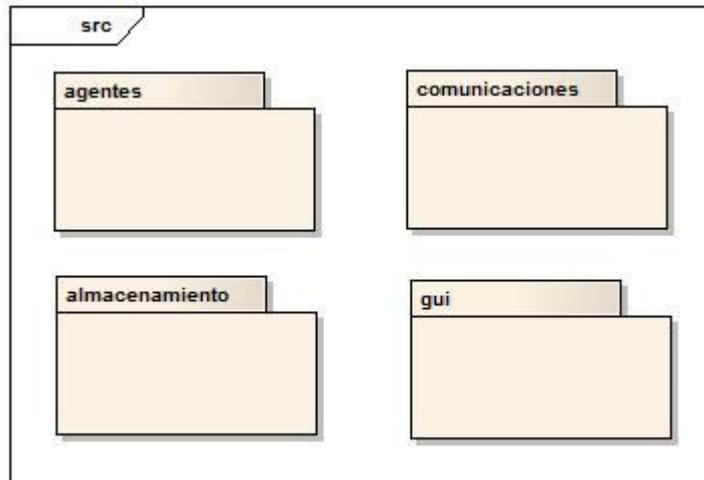


Ilustración 69 - Diagrama src

A continuación se muestran los diagramas de clases correspondientes a cada uno de los paquetes, de forma separada.

El primero de ellos es el paquete que contiene las clases que forman la interfaz de usuario. Como puede verse, incluye las ventanas principales así como las ventanas emergentes y una clase auxiliar *JMap.java* que fue implementada para poder mostrar el mapa de situación del usuario.

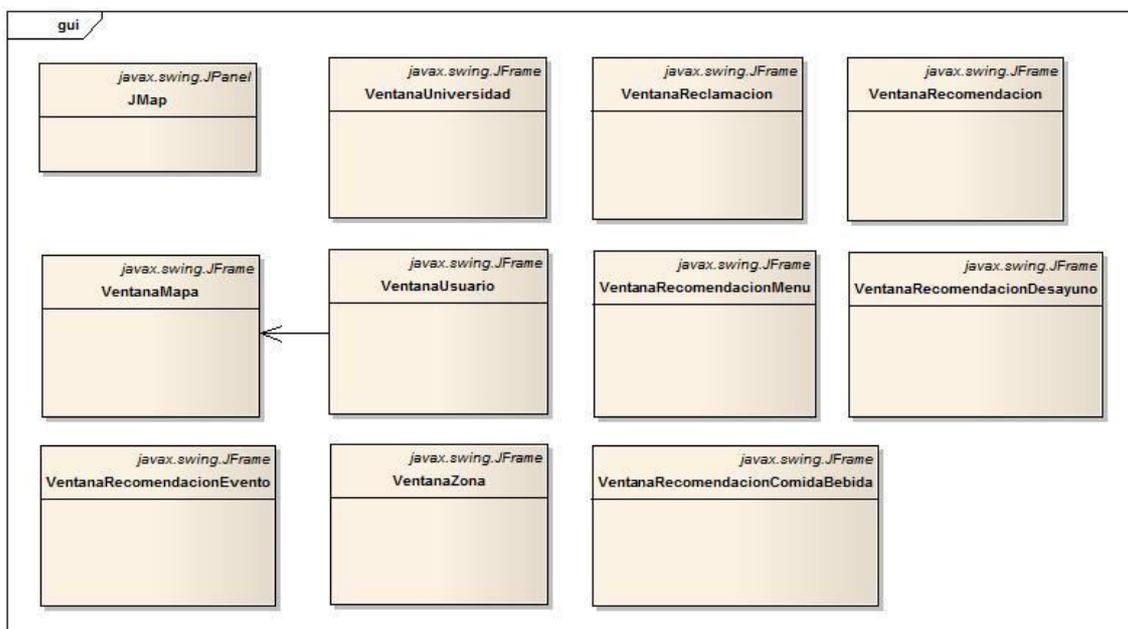


Ilustración 70 - Diagrama gui

El paquete de almacenamiento únicamente contiene una clase cuya funcionalidad es la de leer y escribir en los archivos XML.

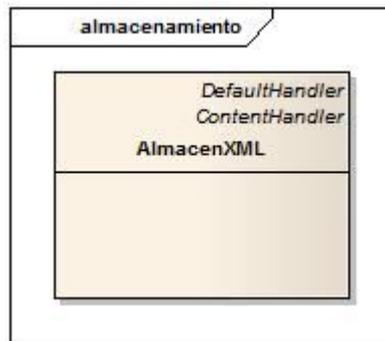


Ilustración 71 - Diagrama almacenamiento

Todo lo relacionado con las comunicaciones entre los agentes se implementa en el paquete del mismo nombre. En este se define el dispositivo de comunicación que los agentes emplean para enviarse mensajes, así como la definición de los mensajes que cada agente puede enviar o recibir y las acciones a llevar a cabo al hacerlo.

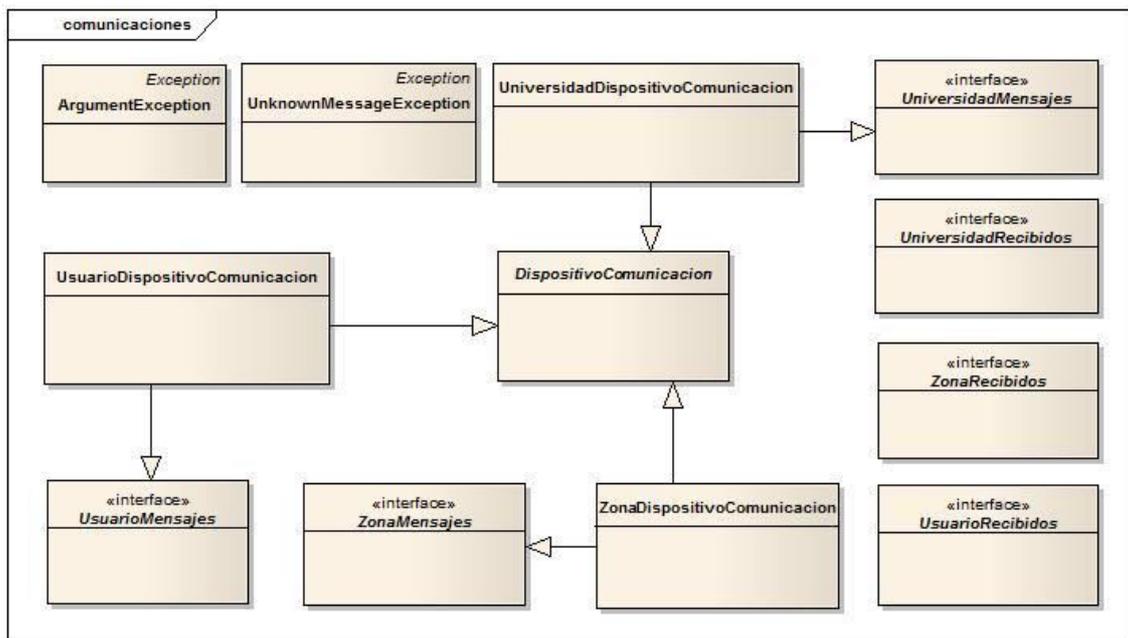


Ilustración 72 - Diagrama comunicaciones

Finalmente, en el paquete agentes se encuentran todas las clases cuya finalidad es la definición de los agentes y de todo lo relacionado con ellos.

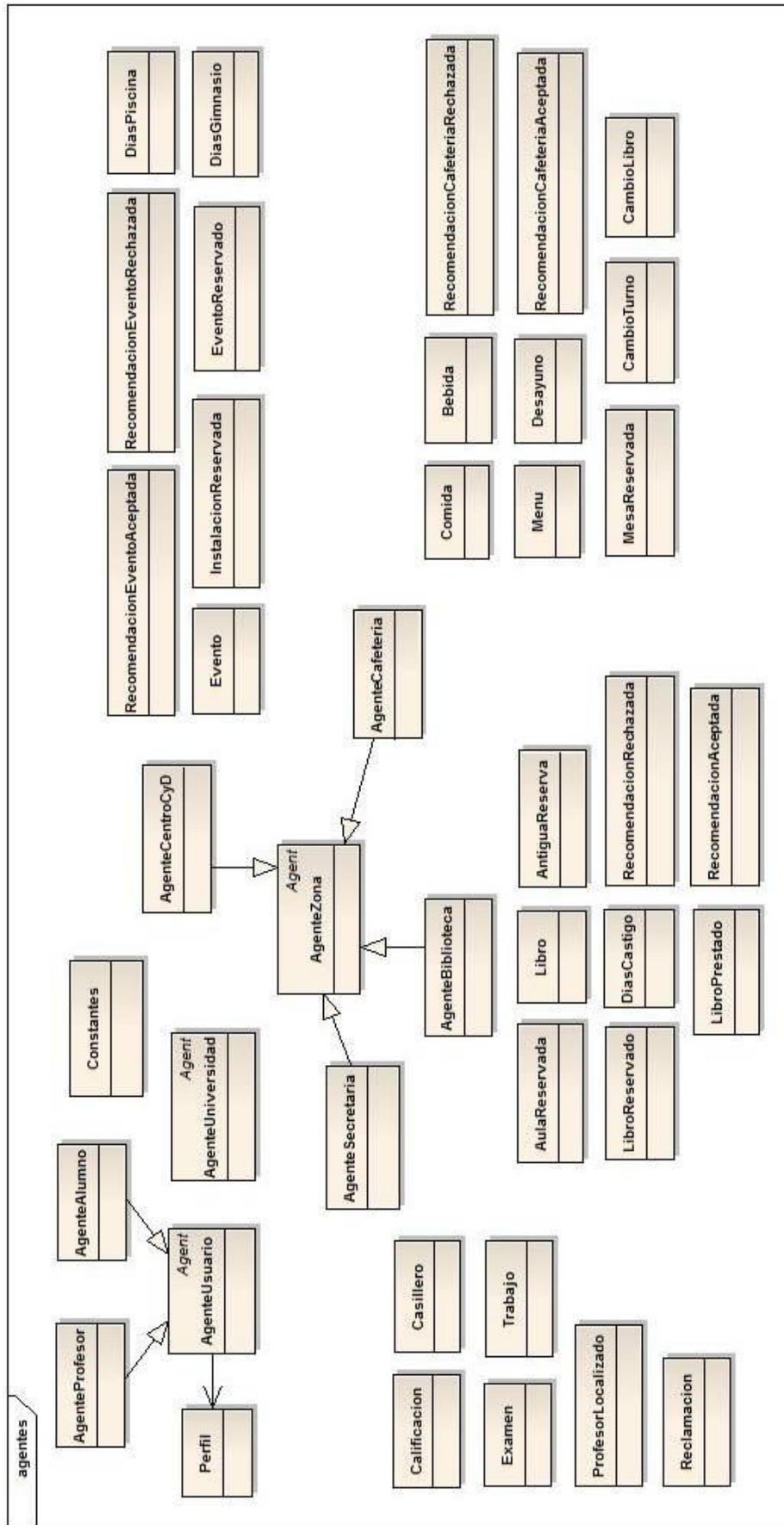


Ilustración 73 - Diagrama agentes

Anexo B – Distribución de la carga de trabajo

A continuación se muestra una tabla donde se calcula una aproximación de las horas dedicadas a cada una de las fases del proyecto. En total, la realización de este ha tenido una duración de 18 meses, alternando algunos meses muy productivos y otros en los que no pudo ser posible avanzar todo lo deseado.

Haciendo un cálculo aproximado y eliminando las temporadas en que apenas se pudo avanzar, la distribución de la carga de trabajo queda como se muestra a continuación.

	Media Horas/Día	Nº Días	Horas Totales
Análisis	2	60	120
Diseño	3	120	360
Desarrollo	4	120	480
		TOTAL	960

Tabla 57 - Distribución de la carga de trabajo

Anexo C – Escenario de uso para la presentación

“David llega a la universidad y conecta el sistema en su teléfono móvil. Como la entrada está al lado la Secretaría lo primero que recibe es un mensaje indicándole que ha obtenido un 9.0 en la asignatura de “Ingeniería de Conocimiento”; el sistema le ofrece poner una reclamación pero como David está satisfecho con su nota decide cancelar el ofrecimiento. Además la Secretaría le recuerda que tiene pendiente la entrega de la práctica de “Programación Avanzada”. A continuación decide irse a tomar algo a la Cafetería. Nada más entrar el sistema le ofrece unas patatas bravas que David acepta muy gustosamente, ya que no le entusiasma el dulce. Además en el tablón de anuncios se almacena su petición de demanda del libro de “Programación Cuarto”.

Quince minutos después, Eva llega a la universidad y conecta el sistema en su PDA. La primera zona por la que pasa todos los días es la de Secretaría y, al entrar en esta, el sistema le comunica que ha salido la nota del examen de “Redes de Ordenadores II” que estaba esperando y que su calificación es de 5.0. El sistema le ofrece la posibilidad de presentar una reclamación y Eva, al haber obtenido una calificación muy baja, la acepta y la envía con su queja correspondiente. Además el agente de la Secretaría le comunica que han encontrado una cartera roja que podría ser la que Eva había perdido hace unos días. Antes de ir a clase Eva acude a la Cafetería en busca de su amigo David y al entrar en dicha zona el agente correspondiente le indica automáticamente que este se encuentra ya allí y, a continuación, le recomienda comer una sopa con filete de pollo y yogurt, aunque en esta ocasión Eva lo rechaza porque ya había comido en casa. Casualmente el sistema indica a Eva que se ha encontrado un alumno para intercambiar con ella el libro de “Programación Cuarto” que ella ofertaba y este es su amigo David. Después de las clases, va a la Biblioteca a devolver un libro, cosa que le es avisada automáticamente nada más entrar en la Biblioteca. Además le llega la notificación de que ha entrado un nuevo libro de Laura Gallego, su autora favorita, que podría interesarle y Eva lo acepta y se lo lleva. Al finalizar el día Eva pasa por el Centro Cultural y Deportivo para recoger unas entradas para un partido de fútbol. Nada más entrar, el sistema le recuerda que lleva 12 días sin pasar por el gimnasio. A cambio de la regañina le recomienda un espectáculo de teatro para el viernes por la noche, que por supuesto acepta para ella y otras 3 personas, volviendo a casa con planes para el fin de semana.

Esa misma tarde, Francisco termina de impartir sus clases y decide pasar por Secretaría como cada tarde para comprobar su casillero. Como es nuevo en la universidad a menudo se desorienta, por lo que consulta el mapa para ver donde se encuentra. Cuando llega a la Secretaría se le comunica que tiene 5 entregas en su casillero, además de la reclamación que Eva le había presentado esa misma mañana. Además se le indica que un profesor al que estaba buscando se encuentra en su despacho en este momento.”

