



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de un sistema multiagente con agentes representativos de modelos de perfiles de usuario de redes sociales

TRABAJO FIN DE MÁSTER

Máster Universitario en Ingeniería Informática

Autor: Daniel Pérez García

Tutor: Estefanía Argente Villaplana

Curso 2019-2020

*Este trabajo habría sido imposible de realizar sin la ayuda de...
mis padres, por una innumerable cantidad de razones,
mi tutora, por su infinita paciencia, trabajo y apoyo constante,
mis compañeros, por su esfuerzo previo que da sentido
a este trabajo, y su asesoramiento.*

Resum

Amb la popularització de les xarxes socials, sobretot entre la població més jove, és cada vegada més freqüent trobar informació personal a la xarxa. Tot i que aquest tipus de plataformes implementen certs mecanismes de seguretat, són pocs els usuaris que inverteixen el temps necessari en aprendre a manejar-los. Aquest treball planteja el desenvolupament d'un sistema multiagent format per agents que representen els diferents models de perfils d'usuari de xarxes socials. Els agents interaccionaran entre si i també amb usuaris reals d'una xarxa social, amb l'objectiu d'aportar-li contingut, simulant els patrons de comportament típics existents en aquest tipus de plataformes. El sistema multiagent s'integrarà a la xarxa social PESEDIA, que té l'objectiu didàctic d'assessorar sobre la privacitat i l'abast de les seues accions als usuaris de la xarxa. En l'actualitat, PESEDIA compte només amb la informació proporcionada pels alumnes de l'Escola d'Estiu de la UPV, en les seues recents edicions. El sistema multiagent permetrà dotar la xarxa de major contingut, fictici però consistent amb el que s'esperava en qualsevol xarxa social, facilitant als desenvolupadors de PESEDIA la validació de les eines implementades a la xarxa social.

Paraules clau: sistema multiagent, agents intel·ligents, patrons d'ús, privacitat, perfils d'usuari, simulació, xarxes socials

Resumen

Con la popularización de las redes sociales, sobre todo entre la población más joven, es cada vez más frecuente encontrar información personal en la red. A pesar de que este tipo de plataformas implementan ciertos mecanismos de seguridad, son pocos los usuarios que invierten el tiempo necesario en aprender a manejarlos. Este trabajo plantea el desarrollo de un sistema multiagente formado por agentes que representan los diferentes modelos de perfiles de usuario de redes sociales. Los agentes interaccionarán entre sí y también con usuarios reales de una red social, con el objetivo de aportarle contenido, simulando los patrones de comportamiento típicos existentes en este tipo de plataformas. El sistema multiagente se integrará en la red social PESEDIA, que tiene el objetivo didáctico de asesorar sobre la privacidad y el alcance de sus acciones a los usuarios de la red. En la actualidad, PESEDIA cuenta solamente con la información proporcionada por los alumnos de la Escola d'Estiu de la UPV, en sus recientes ediciones. El sistema multiagente permitirá dotar a la red de mayor contenido, ficticio pero consistente con lo esperado en cualquier red social, facilitando a los desarrolladores de PESEDIA la validación de las herramientas implementadas en la red social.

Palabras clave: sistema multiagente, agentes inteligentes, patrones de uso, privacidad, perfiles de usuario, simulación, redes sociales

Abstract

With the popularization of social networks, especially among the younger population, it is increasingly common to find personal information on the web. Despite the fact that this type of platform implements certain security mechanisms, few users invest the necessary time to learn how to use them. This work proposes the development of a multi-agent system formed by agents that represent the different models of user profiles within social networks. The agents will interact with each other and also with real users of a social network, with the aim of providing content, simulating the typical behavior patterns existing in this type of platform. The multi-agent system will be integrated into

the PESEDIA social network, which has the didactic objective of advising network users on privacy and the scope of their actions. At present, PESEDIA has only the information provided by the students of the Escola d'Estiu de la UPV, in its recent editions. The multi-agent system will provide the network with more content, fictitious but consistent with what is expected in any social network, making it easier for PESEDIA developers to validate the tools implemented in the social network.

Key words: multi-agent system, intelligent agents, usage patterns, privacy, user profiles, simulation, social networks

Índice general

Índice general	VII
Índice de figuras	IX
Índice de tablas	XI
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Impacto esperado	2
1.4 Metodología	3
1.5 Estructura	3
2 Estado del arte	5
2.1 PESEDIA: Explicación, motivaciones y objetivos	5
2.2 Trabajos relacionados	8
2.2.1 Jarvee	9
2.2.2 FollowingLikeII	10
2.2.3 FollowLiker	11
2.2.4 Somiibo	12
2.2.5 Social10x	13
2.2.6 Conclusiones sobre trabajos relacionados	14
2.3 Plataformas de sistemas multiagente	16
2.3.1 JADE	17
2.3.2 Jadex	18
2.3.3 JACK	19
2.3.4 JASON	20
2.3.5 SPADE	21
2.3.6 Conclusiones	22
2.4 Perfiles de usuario	23
2.4.1 Personalidad, edad y sexo	24
2.4.2 Privacidad y experiencia	28
2.4.3 Hábitos y patrones de uso en redes sociales	33
2.5 Conclusiones	36
3 Análisis del problema	37
3.1 Especificación de requisitos	37
3.1.1 Introducción	37
3.1.2 Descripción general	39
3.1.3 Requisitos específicos	43
3.2 Análisis del marco legal y ético	46
3.3 Análisis de riesgos	47
3.4 Identificación y análisis de soluciones posibles	48
3.5 Conclusiones	49
4 Solución propuesta	51
4.1 Especificación conceptual	51

4.1.1	Visión global	51
4.1.2	Comunicaciones entre los subsistemas	54
4.1.3	Esquema de red	58
4.2	Especificación formal	60
4.2.1	Capa de presentación	60
4.2.2	Capa de negocio	62
4.2.3	Capa de persistencia	64
4.3	Modelos de los agentes	65
4.3.1	Factores	65
4.3.2	Acciones	66
4.3.3	Modelos de ejemplo	67
4.4	Plan de trabajo	68
4.5	Presupuesto	70
4.6	Conclusiones	71
5	Implementación y validación	73
5.1	Implementación	73
5.1.1	Hito 0 - Preparación de la infraestructura	73
5.1.2	Hito 1 - Mecanismos de comunicación	75
5.1.3	Hito 2 - Creación de nuevos agentes y usuarios	78
5.1.4	Hito 3 - Creación de círculos de amigos dentro del sistema multi- agente	79
5.2	Validaciones	83
5.2.1	Validación del Hito 1	83
5.2.2	Validación del Hito 2	85
5.2.3	Validación del Hito 3	86
5.2.4	Pruebas de rendimiento	88
5.3	Conclusiones	89
6	Conclusiones	91
6.1	Relación del trabajo desarrollado con los estudios cursados	92
6.2	Trabajos futuros	94
	Bibliografía	97

Índice de figuras

2.1	Experimento realizado durante la Escola d'Estiu 2019 con el objetivo de comprobar la facilidad que tienen los niños para comprender emociones. . .	6
2.2	Vistas con la información del perfil de usuario en PESEDIA y la configuración de perfil, en las que se puede observar que en todo momento el usuario puede definir el alcance de su información y sus publicaciones. . .	7
2.3	En la primera imagen se muestra el tipo de mensaje que verá un usuario en PESEDIA cuando los agentes detectan que el contenido que se ha publicado puede tener un factor de riesgo, en este caso emocional. En la segunda imagen se muestra un diálogo que se ofrece al usuario cuando una publicación contiene información personal, en este caso de localización.	8
2.4	Dashboard de Jarvee	9
2.5	Dashboard de FollowingLikeII y opciones de programación de una nueva tarea	10
2.6	Dashboard de FollowLiker y ventana de selección de acciones	11
2.7	Dashboard de Somiibo	13
2.8	Servicios ofertados por Social10x para YouTube.	14
2.9	Popularidad de los trabajos relacionados estudiados basado en datos extraídos de Google Trends. El valor 100 refleja el término con mayor popularidad (Jarvee) y el resto de datos muestran la popularidad en comparación con ese término, indicando 50 que el término es la mitad de popular, 0 que no es nada relevante, etc.	15
2.10	Tendencia de popularidad de las plataformas estudiadas basado en datos extraídos de Google Trends. El valor 100 refleja el término con mayor popularidad (Jadex) y el resto de datos muestran la popularidad en comparación con ese término, indicando 50 que el término es la mitad de popular, 0 que no es nada relevante, etc.	16
2.11	Arquitectura de JADE	17
2.12	GUI de JADE	17
2.13	Arquitectura de Jadex	18
2.14	Jadex Control Center	18
2.15	Jack IDE	19
2.16	Modificación de JEdit para trabajar con JASON	20
2.17	Consola de JASON. En la imagen se puede observar la interfaz gráfica de un programa en ejecución junto al detalle de las acciones que se han ejecutado	21
2.18	GUI de SPADE	22
2.19	Comparativa de popularidad media de las herramientas analizadas basado en datos extraídos de Google Trends	23
2.20	Modelo del Big Five	24
2.21	Clustering de los datasets empleados por Gerlach et al. para extraer y proponer sus perfiles de personalidad en base a los factores del Big Five. Los símbolos utilizados sirven para diferenciar los cuatro datasets empleados en el estudio. Fuente [23]	26

2.22	Correlación entre sexo, edad y perfiles propuestos por Gerlach et al. En la imagen A se puede observar la distribución por edades y sexos en los diferentes datasets. En la figura B se observa su frecuencia dentro de los cuatro perfiles, siendo las muestras poco representativas indicadas con fondo en gris. Fuente [23]	27
2.23	Relación entre acciones y preocupación para preservar la privacidad. Fuente [46]	29
2.24	Relación entre perfiles de privacidad y acciones realizadas para preservarla. Fuente [51]	30
2.25	Relación entre perfiles de experiencia gestionando opciones de privacidad y acciones/opciones reconocidas por el usuario para preservarla. Fuente [51]	31
2.26	Relación entre perfiles de privacidad y niveles de experiencia de los usuarios. Entre paréntesis se indica la diferencia respecto a la media en la relación entre grupos, también apoyado por el color de la celda. Fuente [51]	32
3.1	Diagrama de casos de uso	40
4.1	Diagrama conceptual del sistema	52
4.2	Mecanismos de comunicación empleados por el sistema	54
4.3	Infraestructura de red actual	59
4.4	Infraestructura de red propuesta	59
4.5	Fotomontaje de la versión web para establecer conexión con el sistema multiagente	60
4.6	Fotomontaje de la versión web para consultar el estado del sistema multiagente y realizar modificaciones	61
4.7	Fotomontaje de la versión móvil	61
4.8	Diagrama correspondiente a la acción de solicitud de estado - CU01	62
4.9	Diagrama correspondiente a la acción de añadir o retirar agentes de la plataforma - CU02	63
4.10	Diagrama correspondiente a la acción de responder y persistir un mensaje por parte de los agentes para su posterior visualización - CU03 y CU04	63
4.11	Diagrama correspondiente a la acción de añadir un amigo por parte de los agentes - CU06	64
4.12	Plan de trabajo usando un diagrama de Gantt	70
5.1	Esquema de comunicación de los contenedores en Docker	75
5.2	Infraestructura Docker utilizada en el desarrollo del proyecto	75
5.3	Esquema de llamadas para responder a peticiones de estado desde la red social y el sistema multiagente	77
5.4	Diagrama de clases de tipos de usuarios y sus comportamientos	82
5.5	Diagrama de secuencia mostrando el proceso de comunicación entre agentes al añadir un amigo	82
5.6	No se puede activar el plugin porque faltan dependencias	83
5.7	Información de dependencias del plugin	83
5.8	Error al activar el plugin cuando el sistema multiagente no está disponible	84
5.9	Ventana de configuración del plugin	84
5.10	Mensaje de error que muestra la API cuando no se utiliza la clave necesaria para hacer consultas	84
5.11	Mensaje que recibirá el sistema multiagente cuando PESEDIA funcione correctamente	85
5.12	Usuarios ficticios creados en PESEDIA	85
5.13	Ventana de depuración del sistema multiagente en el que se puede ver la lista de agentes creados en el sistema	86

5.14	Ventana de configuración del plugin con información de agentes	86
5.15	Evolución del deseo por añadir amigos que tendrán los agentes	87
5.16	Número de amigos que tienen los agentes al finalizar la ejecución, usando un reparto de 50-50 según su tipología	88
5.17	Red de amigos generada tras una ejecución con 12 agentes	88
5.18	Análisis del impacto en memoria del sistema multiagente	89

Índice de tablas

2.1	Acciones soportadas por Javee en las redes sociales más relevantes	9
2.2	Acciones soportadas por FollowingLikeII en las redes sociales más relevantes	11
2.3	Acciones soportadas por FollowLiker en las redes sociales más relevantes	12
2.4	Acciones soportadas por Somiibo en las redes sociales más relevantes	12
2.5	Servicios disponibles para adquisición en Social10x en las redes sociales más relevantes	13
2.6	Comparativa de trabajos relacionados	15
2.7	Comparativa de plataformas de sistemas multiagente	23
3.1	Descripción de los casos de uso	41
3.2	Tipos de usuarios y sus características	41
3.3	Restricciones del sistema	42
3.4	Requisito S1-RI01	43
3.5	Requisito S1-RI02	43
3.6	Requisito S1-RF01	44
3.7	Requisito S1-RF02	44
3.8	Requisito S1-RF03	44
3.9	Requisito S1-RF04	44
3.10	Requisito S1-RF05	44
3.11	Requisito S1-RF06	44
3.12	Requisito S1-RF07	45
3.13	Requisito RR01	45
3.14	Requisito RR02	45
3.15	Requisito RR03	45
3.16	Requisito RR04	45
3.17	Requisito RD01	46
3.18	Riesgo RUsu01	47
3.19	Riesgo RUsu02	48
3.20	Riesgo RInt01	48
3.21	Riesgo RInt02	48
3.22	Riesgo RInt03	48
3.23	Riesgo RInt04	48
4.1	Factores que definen los modelos de los agentes, con sus posibles valores .	65
4.2	Factores que definen las acciones que podrán realizar los agentes, con sus posibles valores según el interés mostrado	67
4.3	Perfiles de usuario y actividad Tipo 1	67
4.4	Perfiles de usuario y actividad Tipo 2	68

CAPÍTULO 1

Introducción

1.1 Motivación

En la última década la popularización de las redes sociales ha propiciado que este tipo de herramientas se conviertan en una parte integral del día a día de sus usuarios permitiéndoles volcar en internet sus quehaceres, pensamientos, opiniones, conflictos, etc. Esta familiaridad ha llevado a sus usuarios a obviar ciertos aspectos en lo relativo a su privacidad, especialmente en las generaciones más jóvenes, que han crecido con ellas y en algunas ocasiones llegan a considerarlas, erróneamente, inocuas [32]. Por este motivo en el año 2014 decidió ponerse en marcha el proyecto nacional “Privacidad en Entornos Sociales Educativos durante la Infancia y la Adolescencia” a manos del Grupo de Tecnología Informática e Inteligencia Artificial (en adelante GTI-IA) del DSIC.

A raíz del mencionado proyecto, posteriormente en el ejercicio 2015/2016 se puso en marcha el desarrollo de PESEDIA [2], una red social propia del GTI-IA, con el objetivo didáctico de conseguir enseñar a sus usuarios una serie de preceptos básicos entorno a la privacidad y el alcance que puede tener su paso por la red, la llamada huella digital. Esta red social viene siendo utilizada con éxito desde entonces como herramienta de investigación en los talleres de las escuelas de verano organizadas por la universidad. Gracias a estos estudios se ha llegado a la conclusión de que existen ciertos patrones de uso comunes a la mayoría de los jóvenes que sería deseable corregir, como la excesiva confianza para añadir como amigos a personas que no conocían físicamente [6].

Con el objetivo de ayudar a corregir dichas malas prácticas se ha considerado necesario el desarrollo de una serie de agentes inteligentes que interactúen entre ellos y con los usuarios simulando el comportamiento de un ser humano para ayudarles a entender la importancia de sus acciones en la red social, de modo que el usuario sea consciente en todo momento de lo que está ocurriendo o podría ocurrir en caso de realizar un mal uso de la herramienta.

1.2 Objetivos

El propósito del proyecto consiste en el desarrollo de una serie de agentes inteligentes que actuarán sobre la red social PESEDIA y que realizarán las siguientes actividades:

- Tras realizar el proceso de configuración de los agentes a través de un plugin incluido en PESEDIA se procederá a la creación de los agentes.

- Una vez creados, estos agentes tratarán de conectarse a la red social utilizando usuarios ya existentes en la plataforma o creando nuevos usuarios siempre que fuese necesario.
- Una vez conectados realizarán todas las acciones necesarias para tratar de cumplir sus objetivos, definidos a través de una serie de modelos teóricos. Los agentes interactuarán entre sí y también con los usuarios reales de la plataforma con el objetivo de simular el comportamiento de un humano.

Para realizar esta implementación será necesario el estudio de diferentes temas:

- En primer lugar se analizarán distintos perfiles de usuarios de cara a plantear la forma en la que los agentes interactuarán con la red social.
- En segundo lugar se analizarán los principales tipos de interacciones que puede realizar un usuario sobre la plataforma PESEDIA con el objetivo de obtener las actividades más relevantes de cara al posterior desarrollo de los agentes.
- Finalmente se determinarán diferentes escenarios de estudio basados en las principales actividades obtenidas previamente con el propósito de, una vez implementados los agentes, comprobar si han cumplido sus objetivos.

Una vez realizada la fase de estudio se procederá a la implementación de los agentes. Para ello será necesario determinar si los agentes actuarán directamente sobre PESEDIA o si se utilizará otro tipo de plataforma que se conecte a PESEDIA y que se encargue de gestionar la interacción, como por ejemplo la plataforma de agentes SPADE. Este punto es clave debido a que en ciertas situaciones es posible que el usuario deba esperar a que los agentes evalúen sus interacciones para permitirle continuar utilizando la aplicación, por lo que se debe realizar énfasis en la velocidad de respuesta de los agentes.

Finalmente se procederá a comprobar si los agentes han cumplido los objetivos planteados en los diferentes escenarios tras dejarles interactuar con la red social el tiempo estimado. También se analizará el impacto computacional que han tenido sobre la plataforma con el propósito de comprobar si en el futuro sería factible tener agentes similares que interactúen con los usuarios en tiempo real con la intención de ayudarles a corregir sus malas prácticas, proporcionándoles consejos basados en sus acciones.

1.3 Impacto esperado

Al tratarse de un proyecto de innovación los objetivos de este trabajo se encuentran alineados en gran medida con el objetivo número 9 (Industria, innovación e infraestructura) de la lista de objetivos de desarrollo sostenible de la ONU. Una vez completado, se espera que el proyecto pueda ayudar a:

- En primer lugar, y más importante, a los potenciales usuarios de la red social, proporcionándoles un entorno seguro en el que aprender los peligros de su paso por ésta u otras plataformas similares.
- En segundo lugar, al equipo de investigación del GTI-IA, proporcionándoles las herramientas necesarias para simular el comportamiento de un humano en la plataforma, eliminando en parte la necesidad de personas físicas para la realización de pruebas.

Este trabajo forma parte integral del proyecto de investigación *Agentes Inteligentes para Asesorar en Privacidad en Redes Sociales (TIN2017-89156-R)*, por lo que se continuará con el mismo en el futuro a través de los diferentes trabajos presentes y futuros que puedan surgir. Uno de esos trabajos, correspondiente a mi proyecto de tesis doctoral financiada por la beca asociada a dicho proyecto, consistirá en dotar a los agentes de la capacidad de "adaptarse" a un entorno cambiante a través de un marco normativo en el que se integren nuevas normas sociales que emerjan en base a las interacciones entre los usuarios y las definidas en la configuración de los distintos agentes.

1.4 Metodología

Para la consecución del proyecto se seguirán los pasos enunciados a continuación:

1. Estudio y selección de herramientas a emplear.
2. Descripción de perfiles de usuario según su personalidad.
3. Descripción de perfiles de usuario según su experiencia y comportamiento en el ámbito de la privacidad.
4. Generación de modelos para representar la "personalidad" de los agentes.
5. Definición de la infraestructura de red necesaria.
6. Desarrollo de las soluciones software necesarias.
7. Fase de pruebas funcionales y de carga.
8. Validación y cierre de proyecto.

1.5 Estructura

Teniendo en cuenta lo visto en los apartados anteriores y con el objetivo de estructurar los pasos seguidos para el desarrollo, la consecución y divulgación de los resultados obtenidos como fruto de este trabajo, el documento se organizará de la siguiente forma:

- **Capítulo 1: Introducción:** Capítulo actual. En este capítulo se hará una breve descripción y se ofrecerá un vistazo general de lo que se permite alcanzar con este trabajo, indicando entre otros la motivación y los objetivos del mismo.
- **Capítulo 2: Estado del arte:** En este capítulo se realizará un análisis de las principales alternativas en el mercado que puedan realizar una función similar a la planteada en este trabajo. Se realizará además un análisis de trabajos relevantes que permitan generar modelos de usuarios y decidir que herramientas se utilizarán.
- **Capítulo 3: Análisis del problema:** En este capítulo se realizarán las especificaciones y consideraciones a tener en cuenta a nivel tecnológico, moral, legal, etc., identificando los posibles riesgos y posibles soluciones que permitan llevar el proyecto a buen puerto.
- **Capítulo 4: Solución propuesta:** En este capítulo se detallarán las consideraciones necesarias a nivel de proyecto para garantizar el éxito del trabajo.

- **Capítulo 5: Implementación y validación:** En este capítulo se detallará el proceso seguido para la consecución a nivel técnico del proyecto y se evaluarán los resultados obtenidos.
- **Capítulo 6: Conclusiones:** En el último capítulo de este trabajo se hará un repaso del proceso de desarrollo del mismo, destacando los éxitos alcanzados y las trabas encontradas y solventadas. Además se hará un análisis de las diferentes asignaturas con posible relación, que hayan ayudado a la hora de la consecución del trabajo. Finalmente se hará una sugerencia de proyectos futuros surgidos gracias a la aportación de este trabajo.

CAPÍTULO 2

Estado del arte

En el mercado existen multitud de herramientas que permiten simular interacciones simples de usuarios con una red social a través de agentes inteligentes popularmente conocidos como "social bots" [22]. A diferencia de los sistemas actuales, en este proyecto se tratará de dotar a una serie de agentes de los mecanismos necesarios para que simulen tener diferentes personalidades, por lo que en este capítulo además de analizar herramientas similares a la solución propuesta, se analizarán diferentes plataformas para desarrollar sistemas multiagente con la intención de decidir cuál utilizar en este proyecto. Para poder entender el motivo de existencia de los agentes, y que lleva a querer dotarles de personalidad, se explicará en mayor detalle qué es PESEDIA, expandiendo alguno de los conceptos ya vistos en el apartado anterior. Asimismo, se introducirán los conocimientos necesarios para entender los patrones de comportamiento de un humano en una red social en base a sus perfiles de personalidad y relación con la privacidad.

2.1 PESEDIA: Explicación, motivaciones y objetivos

Como ya se ha comentado previamente, PESEDIA es una red social basada en modificaciones sobre la plataforma de código libre Elgg, de desarrollo propio y de carácter educativo, que tiene como principal objetivo enseñar a sus usuarios la importancia de su información personal y los peligros que conlleva no gestionar correctamente el alcance del contenido que se publica en la red. Con esto se pretende dotar a los usuarios de PESEDIA de los conocimientos necesarios para poder hacer un uso responsable de cualquier tipo de red social o plataforma de contenido, incluso aunque dispongan de una gestión y manejo de opciones más complejo que el que ofrece PESEDIA.

A pesar de no importar la edad a la hora de hacer uso de PESEDIA, la población objetivo a la que se enfoca son los adolescentes, al ser el sector que se encuentra más desprotegido ante los peligros de internet y que a su vez mayor uso hace de este tipo de herramientas.

Para conseguir cumplir el objetivo de educar a los jóvenes, se han organizado durante los últimos años una serie de jornadas en la Escola d'Estiu de la UPV en las que, a través de técnicas de gamificación como la mostrada en la figura 2.1, se logra no solamente que los niños aprendan a utilizar la red social de manera correcta, sino que además comprendan la implicación e impacto que tiene cada interacción que realizan. En estas jornadas se aprovecha también para recolectar información anónima que pueda resultar de utilidad de cara a generar modelos que ayuden a comprender la relación que tienen los niños con temas relacionados con la privacidad, su estado emocional y cómo influye éste a la hora de generar contenido, cómo modifican sus patrones de comportamiento según aumenta el conocimiento que tienen de la plataforma, etc.

Los miembros de la red social disponen de las herramientas necesarias para limitar al máximo la información que proporcionan, como puede observarse en la figura 2.2, por lo que en todo momento queda a su elección cómo quieren que se difunda la información, sin olvidar que están dentro de un entorno controlado y seguro. Adicionalmente a esas medidas, los usuarios que se utilizan en la red social están creados previamente y se entregan a las personas que van a tener acceso a la red, con el objetivo de evitar que de manera involuntaria proporcionen su información personal, por lo que todos los datos que se pueden explotar posteriormente son anónimos.

En la actualidad la red social PESEDIA dispone de una serie de agentes que se encargan de asesorar y avisar a los niños cuando van a publicar contenido sensible, a través de análisis de textos, emociones y técnicas de argumentación. En la figura 2.3 se puede observar un ejemplo de las diferentes alertas empleadas para notificar a los niños que sus mensajes contienen información sensible. Estos agentes tienen el objetivo de ayudar a los usuarios a corregir el contenido que publican en las redes sociales, realizando un seguimiento individualizado para lograr una mejora en el tiempo.

Lo que se pretende en este proyecto es conseguir implementar unos agentes que interactúen en la red social de manera orgánica entre ellos y con los usuarios humanos, con el objetivo futuro de dotarles de las herramientas necesarias para que puedan apoyar al usuario con la misma eficacia que los agentes disponibles en la actualidad, pero de manera menos intrusiva, utilizando técnicas paternalistas a través de las herramientas disponibles en la red social como comentarios, "me gusta", mensajes privados, etc., de demostrada eficacia [51].

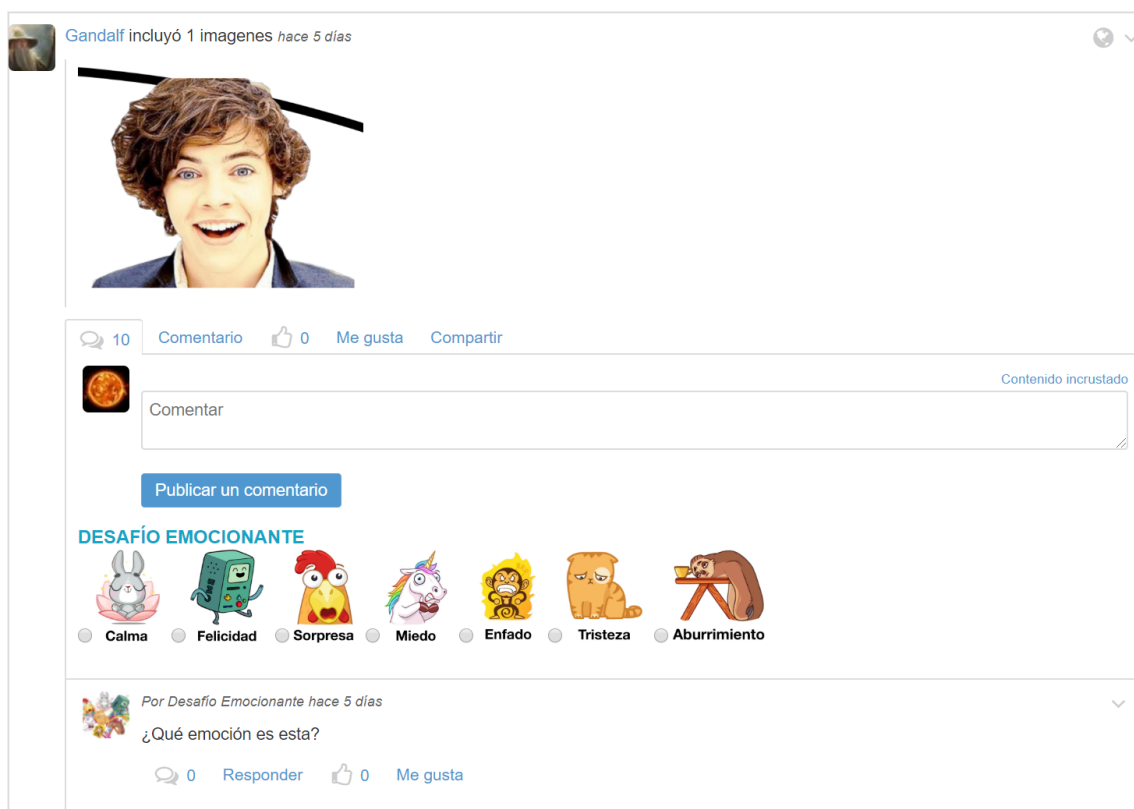


Figura 2.1: Experimento realizado durante la Escola d'Estiu 2019 con el objetivo de comprobar la facilidad que tienen los niños para comprender emociones.

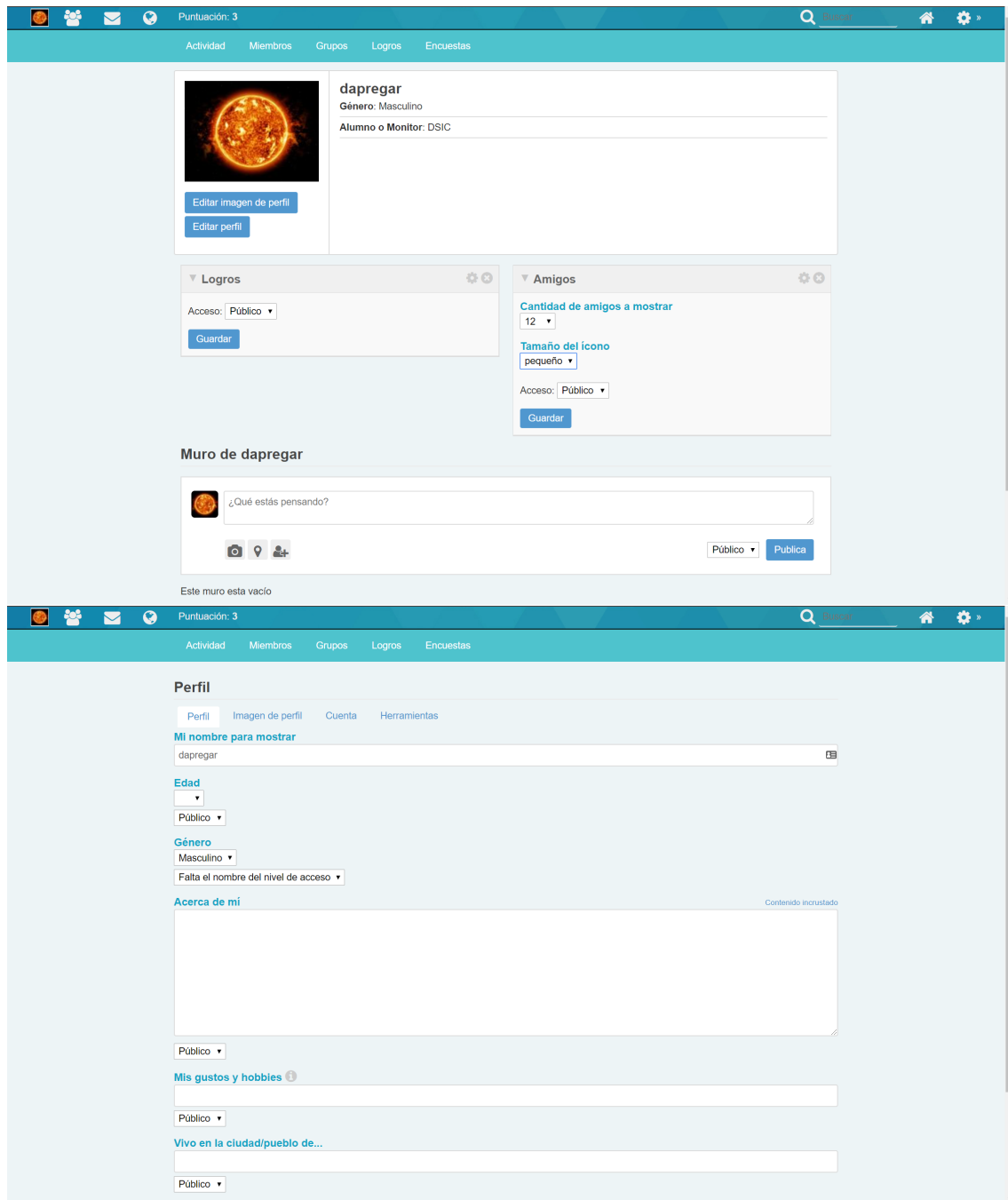


Figura 2.2: Vistas con la información del perfil de usuario en PESEDIA y la configuración de perfil, en las que se puede observar que en todo momento el usuario puede definir el alcance de su información y sus publicaciones.



Figura 2.3: En la primera imagen se muestra el tipo de mensaje que verá un usuario en PESEDIA cuando los agentes detectan que el contenido que se ha publicado puede tener un factor de riesgo, en este caso emocional. En la segunda imagen se muestra un diálogo que se ofrece al usuario cuando una publicación contiene información personal, en este caso de localización.

2.2 Trabajos relacionados

Como ya se ha mencionado existe un gran número de alternativas dentro del campo de la automatización de acciones en las redes sociales. Las más populares, que son las que se analizan en este trabajo, son de propósito general, o lo que es lo mismo, tratan de funcionar en la mayor cantidad de redes sociales posibles, aunque evidentemente también existen alternativas muy populares que tratan de cubrir casos de uso específicos para una única red social.

Sería deseable que este análisis se realizase sobre herramientas que trataran de simular el comportamiento de un humano, pero al tratarse de un campo sobre el que todavía se está teorizando [10] no se han encontrado herramientas que cumplan este propósito.

Aunque el objetivo final de este trabajo no es la implementación de un "bot" que simplemente automatice acciones, como hacen las herramientas que se analizarán en este punto, y en este proyecto solo se tendrán en cuenta las acciones realizables en la red social PESEDIA, a diferencia de las herramientas analizadas que están pensadas para funcionar en diferentes plataformas, es importante conocer las alternativas existentes en el mercado para tener una visión clara de algunos de los diferentes elementos y acciones que podrá contener y realizar nuestra aplicación, así como posibles planes de monetización si los hubiese.

En los siguientes apartados se describen las cinco herramientas más relevantes que implementan "social bots", en base a una entrada publicada en el blog de la página web outbound.net¹ en la que se analizan las que son, en su opinión, las mejores soluciones disponibles en el mercado y contrastando los resultados con el buscador de tendencias de búsquedas en Google, Google Trends², utilizando como referencia los datos de bús-

¹<https://outbound.net/top-social-media-bots-and-automation-tools/>

²<https://trends.google.es/>

queda a nivel mundial, con el objetivo de conocer cuáles son las más populares. Para cada aplicación se adjunta una tabla en la que se indican las principales redes sociales y las labores primordiales que el cliente podría necesitar automatizar o adquirir para que resulte más cómodo comparar los servicios que se ofrecen en cada una de ellas.

2.2.1. Jarvee

Jarvee ³ es un bot diseñado para trabajar sobre equipos con sistema operativo Windows y preparado para trabajar con un gran número de redes sociales como Instagram, Facebook o Twitter. El objetivo principal de Jarvee es la automatización de actividades tediosas para evitar la necesidad de gestionar a mano y repetir acciones en todos los perfiles con los que el usuario necesite trabajar. En la tabla 2.1 se pueden encontrar las acciones disponibles de manera habitual en la mayoría de redes sociales, junto al soporte que ofrece Jarvee de las mismas.

	Gestión publicaciones	Redifusión mensajes populares	Gestión amigos, seguidores y grupos	Gestión mensajes directos	Auto "Me gusta"	Difusión mensajes personalizado	Búsqueda mejores etiquetas
Facebook	✓				✓		
Instagram	✓	✓	✓	✓	✓	✓	✓
LinkedIn	✓		✓	✓	✓	✓	
Pinterest	✓	✓				✓	✓
Tumblr							
Twitter	✓	✓	✓	✓	✓	✓	✓
YouTube	✓		✓		✓	✓	

Tabla 2.1: Acciones soportadas por Jarvee en las redes sociales más relevantes

La herramienta ofrece varios paquetes dentro de su servicio de suscripción mensual que proporcionan acceso a la totalidad de las funcionalidades pero que varían entre ellas en el número de cuentas que permite registrar. Además ofrece un periodo de prueba de 5 días para que los potenciales clientes puedan decidir si les convence.

The screenshot shows the Jarvee 2.1.7.0 dashboard. The main content area is a table with columns for DATE, STATUS, ACCOUNT, and ERROR. The table lists various activities such as 'Jarvee is now active', 'Redirected to paypal to renew subscription.', 'Invite Friends - Facebook Account 1 is active', and 'Finalized group synchronization for the Facebook Account 1 account.'. The status for most activities is 'Jarvee is now active', while one entry for 'Executing a Login operation for the Facebook Account 1 account.' shows an error status of 'YES'. The dashboard also includes a sidebar with navigation options like DASHBOARD, SOCIAL PROFILES, TOOLS, GLOBAL TOOLS, SETTINGS, PROXY MANAGER, MY ACCOUNT, DESTINATION LISTS, and CAMPAIGNS. At the bottom, there are options to manage columns, copy errors to clipboard, show only errors, and clear errors.

Figura 2.4: Dashboard de Jarvee

³<https://jarvee.com/>

Una de las principales debilidades de esta herramienta frente a otras alternativas es su precio, aunque a cambio ofrece una interfaz simple y moderna (ver figura 2.4) y una gran cantidad de tareas programadas. Otra debilidad importantes es que, a fecha de escritura del documento, Jarvee no permite al usuario programar acciones personalizadas, lo que puede provocar situaciones en las que el usuario necesite más de lo que la herramienta le puede ofrecer.

2.2.2. FollowingLikeII

FollowingLikeII⁴, comúnmente conocida como FollowingLike en referencia a su nombre original, es un bot programado para funcionar en diferentes redes sociales que, de igual manera que Jarvee, solamente funciona en ordenadores con sistema operativo Windows, pero la principal diferencia con dicho bot es que, además de disponer de acciones complejas ya programadas, permite crear flujos de trabajo propios a través de acciones simples.

FollowingLikeII ofrece una gran variedad de opciones de pago que permiten tanto la adquisición de una licencia permanente como una serie de suscripciones mensuales y anuales que se diferencian entre ellas según el número de cuentas que podrá gestionar la aplicación y la red social a la que se conectarán. La existencia de tantos planes de pago, aunque beneficiosa para el usuario porque permite reducir el precio en comparación con otras opciones del mercado, puede resultar confusa a primera vista, lo que podría derivar en que potenciales clientes terminen adquiriendo otras soluciones. Por otro lado, esta herramienta no ofrece un periodo de prueba gratuito, aunque ofrece la posibilidad de realizar una suscripción de 3 días a precio reducido.

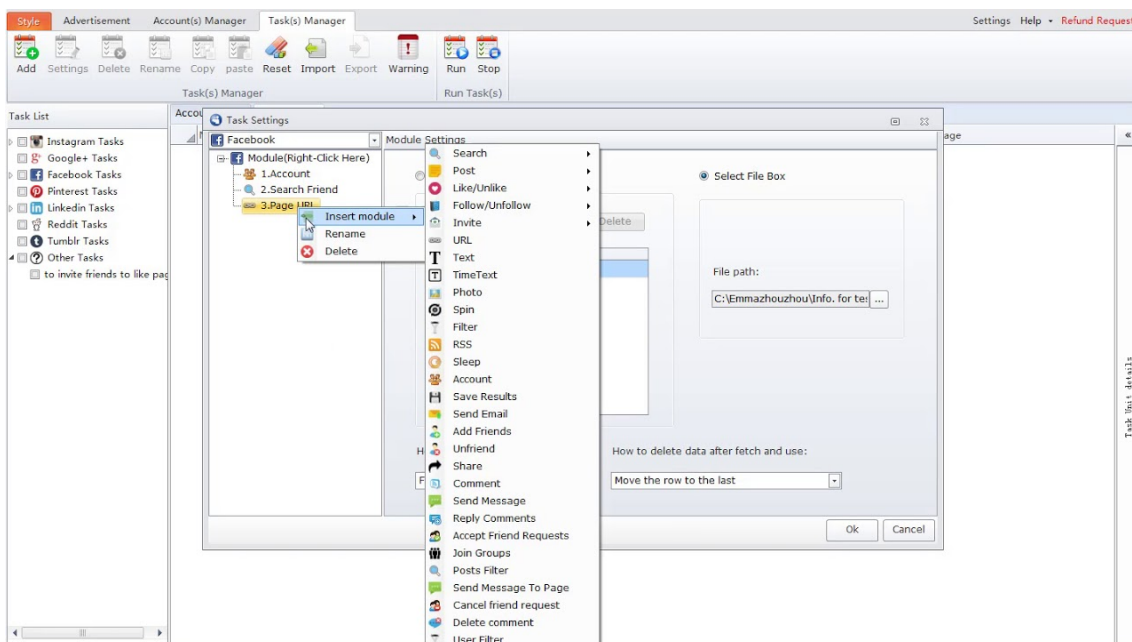


Figura 2.5: Dashboard de FollowingLikeII y opciones de programación de una nueva tarea

Las principales ventajas de esta alternativa son, por un lado, su interfaz (ver figura 2.5), muy similar a las mostradas en otras aplicaciones populares de Microsoft, condensando las opciones de la herramienta en el panel (Ribbon) superior, lo que hará que los usuarios más noveles se puedan familiarizar rápidamente con la aplicación al ser similar a otras que hayan usado previamente y, en segundo lugar, la posibilidad de programar

⁴<http://www.followinglike.com/>

acciones complejas personalizadas, ampliando la funcionalidad de las acciones base del programa (ver tabla 2.2) en base a acciones simples. Por contrapartida, y como ya se ha mencionado, los numerosos planes de pago y la no existencia de un periodo de prueba gratuito pueden resultar un obstáculo de cara a encontrar clientes.

	Gestión publicaciones	Redifusión mensajes populares	Gestión amigos, seguidores y grupos	Gestión mensajes directos	Auto "Me gusta"	Difusión mensajes personalizado	Búsqueda mejores etiquetas
Facebook	✓	✓	✓	✓	✓		
Instagram	✓	✓	✓	✓	✓		
LinkedIn	✓	✓	✓		✓		
Pinterest			✓	✓	✓		
Tumblr							
Twitter							
YouTube							

Tabla 2.2: Acciones soportadas por FollowingLikeII en las redes sociales más relevantes

2.2.3. FollowLiker

FollowLiker⁵ es una herramienta que, de manera similar a las ya analizadas, trata de automatizar tareas repetitivas en varias redes sociales a través de su propio cliente compatible con el sistema operativo Windows. FollowLiker, al igual que Jarvee, no permite al usuario definir sus propios flujos de trabajo, por lo que el usuario queda limitado a las acciones preprogramadas.

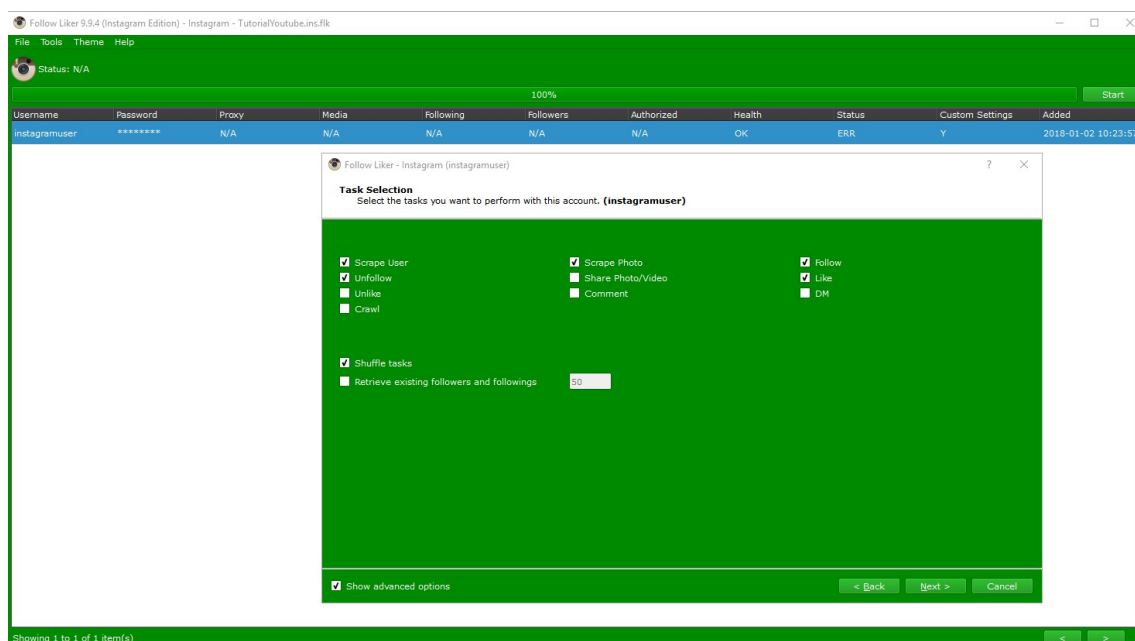


Figura 2.6: Dashboard de FollowLiker y ventana de selección de acciones

Este programa ofrece diferentes versiones y formas de pago, según la red social que se vaya a utilizar, incluyendo: Twitter, Instagram (ver la figura 2.6), Pinterest y Tumblr. El cliente puede optar por comprar una versión que solamente le dé acceso a una red social, una segunda que le dará acceso a dos redes sociales de entre las combinaciones disponibles y una tercera que proporciona acceso a la totalidad de funcionalidad que ofrece la herramienta. La modalidad de pago de las mismas es de pago adquisición, que proporciona acceso a la herramienta durante un mes y, posteriormente, suscripciones mensuales.

⁵<http://www.followliker.com/>

La principal ventaja de esta herramienta es la sencillez de uso. El cliente simplemente tiene que descargar la aplicación, iniciar sesión con la cuenta de la red social sobre la que desee automatizar acciones y seleccionarlas en una ventana simple. Como contrapartida, la simpleza implica que la herramienta carezca de funcionalidades que sí tienen las aplicaciones de la competencia, aunque sigue ofreciendo todas las actividades necesarias que son requeridas habitualmente (ver tabla 2.3). Destaca también la falta de soporte para Facebook, ofertado por el resto de herramientas analizadas. Finalmente esta herramienta es la que tiene los planes de gasto menos favorables para el consumidor, obligándole a hacer un desembolso inicial considerable para adquirir la aplicación y sumando después pagos recurrentes, siendo además un gran problema la falta de una versión de prueba.

	Gestión publicaciones	Redifusión mensajes populares	Gestión amigos, seguidores y grupos	Gestión mensajes directos	Auto "Me gusta"	Difusión mensajes personalizado	Búsqueda mejores etiquetas
Facebook							
Instagram	✓	✓	✓	✓	✓		
LinkedIn							
Pinterest	✓	✓	✓	✓	✓		
Tumblr	✓	✓	✓	✓	✓		
Twitter	✓	✓	✓	✓	✓		
YouTube							

Tabla 2.3: Acciones soportadas por FollowLiker en las redes sociales más relevantes

2.2.4. Somiibo

Somiibo ⁶ es una herramienta que trata de ofrecer multitud de funcionalidades, desde bots que permiten automatizar tareas habituales en distintas redes sociales, tal y como se muestra en la tabla 2.4, pasando por sistemas multiagente para mejorar el posicionamiento SEO en internet hasta acceso a servidores proxy y gestión de tráfico web.

	Gestión publicaciones	Redifusión mensajes populares	Gestión amigos, seguidores y grupos	Gestión mensajes directos	Auto "Me gusta"	Difusión mensajes personalizado	Búsqueda mejores etiquetas
Facebook	✓	✓	✓	✓	✓		
Instagram	✓	✓	✓	✓	✓		
LinkedIn							
Pinterest							
Tumblr							
Twitter	✓	✓	✓	✓	✓		
YouTube							

Tabla 2.4: Acciones soportadas por Somiibo en las redes sociales más relevantes

Esta plataforma ofrece tres opciones de pago diferentes para poder hacer uso de la herramienta. La primera de ellas es una opción gratuita con capacidades reducidas y que impiden el acceso a la mayor parte de opciones de la aplicación, pero que puede resultar útil a los usuarios que simplemente deseen probarla o no necesiten hacer uso de los módulos de pago. Por otro lado se ofrecen dos opciones completamente idénticas que proporcionan acceso total a la aplicación y que tienen modalidad de cobro tanto mensual como anual.

El punto más novedoso y atractivo de esta alternativa es la posibilidad de obtener un amplio abanico de herramientas en una sola aplicación, junto a sus precios, muy competitivos respecto al resto de opciones analizadas, sin dejar de ofrecer una herramienta de aspecto profesional y uso sencillo, tal y como se muestra en la figura 2.7. Como principal defecto destaca la falta de compatibilidad con algunas de las redes sociales más populares, como por ejemplo LinkedIn o Twitter y la imposibilidad de automatizar tareas personalizadas.

⁶<https://somiibo.com/>

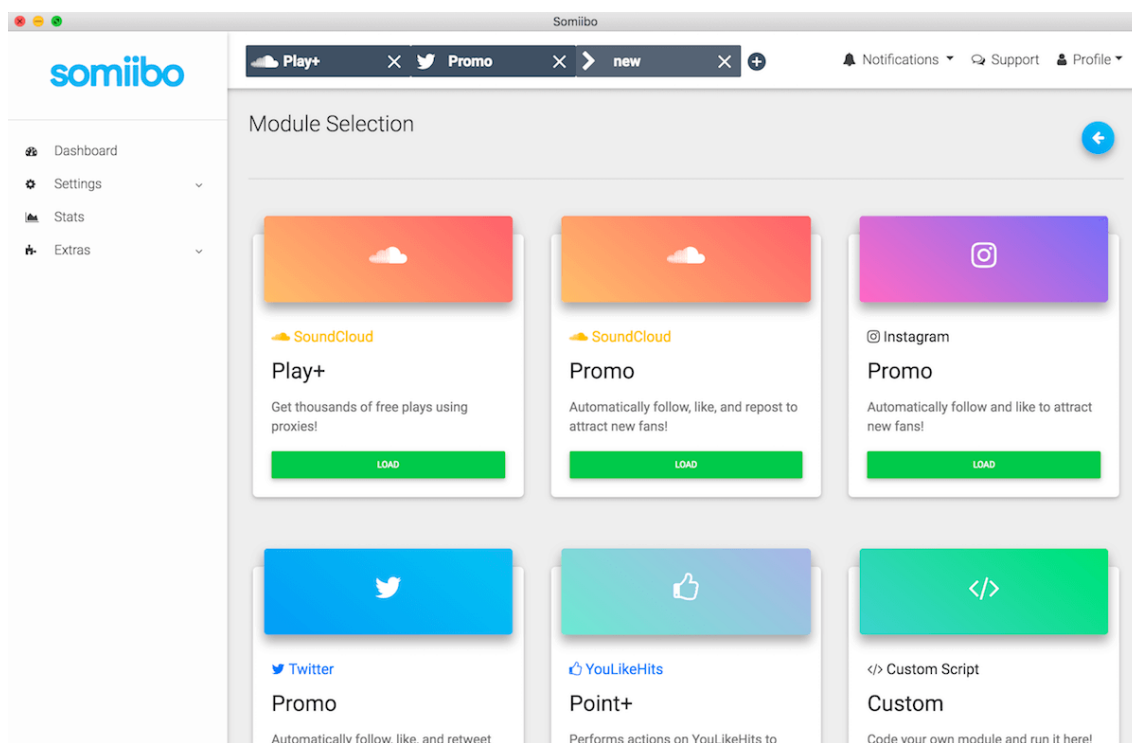


Figura 2.7: Dashboard de Somiibo

2.2.5. Social10x

Social10x ⁷ ofrece un servicio completamente distinto al mostrado por las anteriores alternativas. Esta herramienta no se centra en automatizar acciones sino en ofrecer a sus clientes la opción de incrementar su exposición en diferentes redes sociales con el objetivo de incrementar su número de seguidores. Para conseguirlo explotan los diferentes mecanismos de posicionamiento de las redes sociales objetivo, utilizando un sistema multiagente que tendrá una complejidad y número de agentes determinados por el usuario en el momento de contratación del servicio. Cada agente corresponderá a un usuario ficticio que realizará la acción contratada por el usuario, como puede ser el acceso a un vídeo de YouTube para aumentar su popularidad, seguir al comprador en Instagram e interactuar con los contenidos que vaya publicando para dar la sensación de que es una cuenta popular y así atraer nuevos usuarios reales, etc.

En la tabla siguiente 2.5 se muestran los diferentes servicios ofertados por Social10x. El formato de esta tabla es diferente debido a que esta plataforma no se basa en posibilidad de automatización, que es la métrica empleada en el resto de tablas.

	Comprar seguidores	Comprar interacciones	Comprar "Me gusta"	Comprar redifusión de contenido	Comprar visualizaciones
Facebook	✓		✓		
Instagram	✓	✓	✓		
Twitter	✓		✓	✓	
YouTube	✓		✓	✓	✓

Tabla 2.5: Servicios disponibles para adquisición en Social10x en las redes sociales más relevantes

Los propietarios de esta herramienta ponen a disposición del cliente multitud de planes de pago único para cada servicio ofertado que, como ya se ha mencionado, varían entre sí según el servicio en cuestión, la red social en la que se desean adquirir dichos servicios y el volumen de interacciones que el usuario desee contratar. Para contratar estos

⁷<https://social10x.com/>

servicios, el cliente simplemente debe conectarse a la página web y seleccionar la opción que le resulte de interés de una manera simple, como se muestra en la figura 2.8.

La principal ventaja de esta alternativa es que, al no tratarse de un software, el usuario paga por un servicio y no tiene la necesidad de aprender a usar la herramienta siempre y cuando su único objetivo sea el obtener visibilidad en la red. Como contrapartida esta herramienta carece de las funcionalidades de automatización que se pueden encontrar en el resto de opciones mostradas en este documento, por lo que su utilidad queda muy reducida y, además, se ha observado que la página web ofrece problemas de disponibilidad. Cabe remarcar que, a pesar de su popularidad, las opiniones mostradas en la red acerca de esta herramienta por algunos de sus clientes son muy negativas⁸.

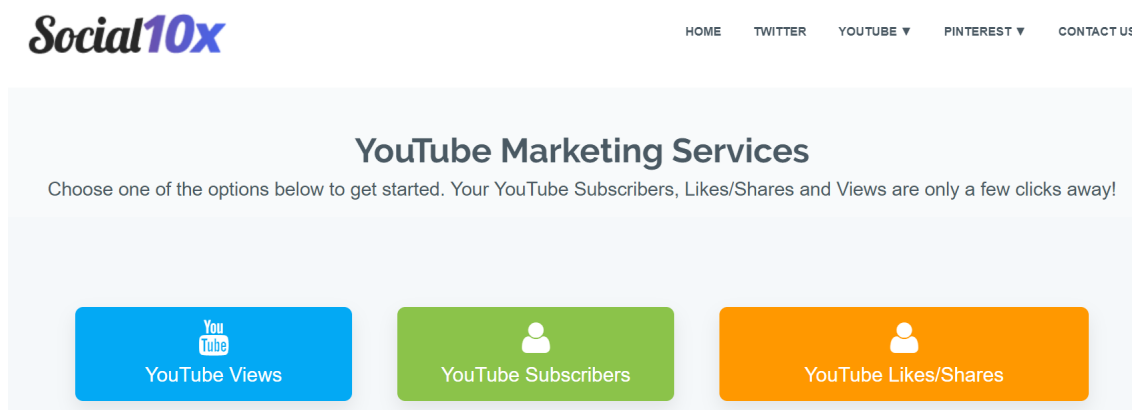


Figura 2.8: Servicios ofertados por Social10x para YouTube.

2.2.6. Conclusiones sobre trabajos relacionados

Tras analizar las propuestas más populares y relevantes del mercado (ver figura 2.9) se han extraído una serie de conclusiones que resultarán de utilidad a la hora de definir ciertas características del producto, sintetizadas en la tabla 2.6:

1. La mayoría de propuestas optan por ofrecer un entorno únicamente funcional en plataformas Windows. Sería un valor añadido diseñar una propuesta que sea funcional en cualquier tipo de entorno, ya sea mediante una plataforma web como haciendo uso de un lenguaje soportado por cualquier sistema operativo.
2. En caso de querer monetizar la aplicación, las opciones más populares pasan por sistemas de suscripciones, quedando el pago por uso/adquisición como una forma poco popular. En caso de optar por un plan de suscripción se debe tratar de simplificar para no abrumar al cliente con opciones.
3. La mayoría de herramientas analizadas no permiten que el usuario defina sus propios flujos de interacción. Puede ser un valor añadido incluir estas opciones pero agrega un nivel de complejidad elevado al desarrollo.
4. La mayoría de alternativas estudiadas ofrecen sincronización con multitud de redes sociales, motivo por el cual en la mayoría de casos no permiten personalizar acciones.

Es importante volver a remarcar que, a pesar de tener numerosos elementos en común con el trabajo que se pretende realizar, ninguno de estos productos busca ofrecer

⁸<https://www.trustpilot.com/review/social10x.com>

las mismas funcionalidades ni características que se esperan conseguir con el producto derivado de este trabajo de fin de máster.

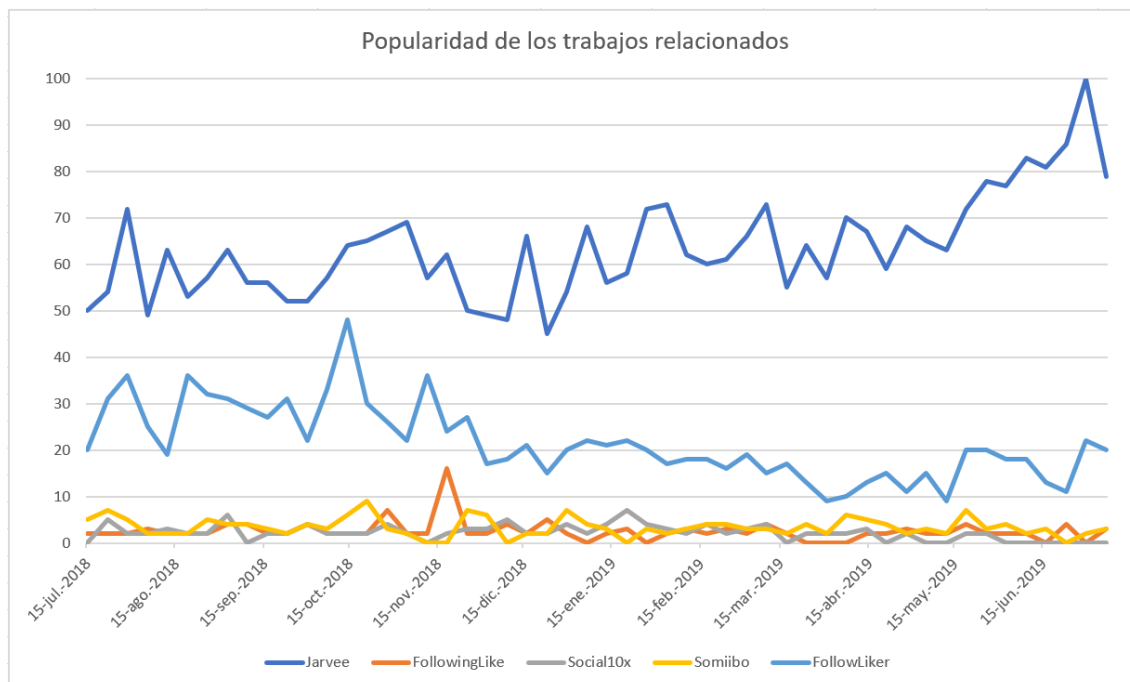


Figura 2.9: Popularidad de los trabajos relacionados estudiados basado en datos extraídos de Google Trends. El valor 100 refleja el término con mayor popularidad (Jarvee) y el resto de datos muestran la popularidad en comparación con ese término, indicando 50 que el término es la mitad de popular, 0 que no es nada relevante, etc.

	Sistemas operativos soportados	Planes de pago	Objetivo	Acciones personalizadas	Redes sociales soportadas
Jarvee	Windows 7 y superiores	Suscripción	Automatizar acciones	No	Facebook, Google+ Instagram, LinkedIn Pinterest, Tumblr Twitter
FollowingLikeII	Windows 7 y superiores	Pago por uso Suscripción mensual Suscripción anual	Automatizar acciones	Sí	Facebook, Google+ Instagram, LinkedIn Pinterest, Reddit Tumblr
FollowLiker	Windows 7 y superiores	Pago por uso + Suscripción mensual	Automatizar acciones	No	Instagram, Pinterest Tumblr, Twitter
Somiibo	Windows 7 y superiores MacOS	Versión gratuita Suscripción mensual Suscripción anual	Automatizar acciones Generar exposición Posicionamiento Proxy web	No	AddMeFast, Facebook Instagram, Sound Cloud Twitter, YouLikeHits
Social10x	Web	Pago por consumo	Generar exposición Posicionamiento	No	Pinterest, Twitter YouTube

Tabla 2.6: Comparativa de trabajos relacionados

Por todos los motivos ya explicados una vez completado el análisis, se considera necesario y justificado el desarrollo de un nuevo producto que se adecúe a las necesidades específicas del proyecto PESEDIA, que posea todas las funcionalidades necesarias, inexistentes en las herramientas analizadas.

2.3 Plataformas de sistemas multiagente

En el mercado puede encontrarse una gran variedad de plataformas que permiten trabajar con sistemas multiagente. Con el objetivo de llevar a cabo el desarrollo de este trabajo, surge la necesidad de realizar un análisis de algunas de estas herramientas para poder determinar cuál de ellas se adecúa mejor a las necesidades que se desean cubrir. Para conseguirlo se ha recurrido a la búsqueda de diferentes estudios comparativos con el objetivo de extraer las características más importantes de cada plataforma, a los que se añadirá información que resulte de interés, siempre que no esté contemplada en los mismos, y que ayudará a decantarse entre una de las opciones analizadas para el desarrollo de nuestros agentes. Las dos primeras propuestas — Kravari, Bassiliades (2015) [28] y Rendón, Sánchez (2006) [42] — realizan un análisis general de las diferentes plataformas para el desarrollo de agentes que hay en el mercado, analizando sus características, protocolos que implementan, cómo de fácil resulta su uso, etc. El tercer artículo revisado, de Donâncio, Casals y Brandão (2019) [19], explica las diferencias a la hora de implementar un sistema multiagente y exponerlo a través de un servicio web, utilizando las plataformas JADE y SPADE.

Para hacer la selección de plataformas a analizar se ha optado por elegir las más populares según la propuesta de Kravari y Bassiliades "A survey of agent platforms" [28] y contrastando la información con Google Trends, obteniendo los resultados mostrados en la figura 2.10. Se ha decidido dar especial relevancia a algunas de las plataformas empleadas en la oferta educativa de la universidad, añadiendo también opciones en las que exista experiencia previa, como SPADE, que aunque tiene popularidad y no está incluida en el citado artículo, es una herramienta desarrollada en el grupo de investigación. Tras este proceso de selección se procede finalmente al análisis de las cinco plataformas que se exponen a continuación.

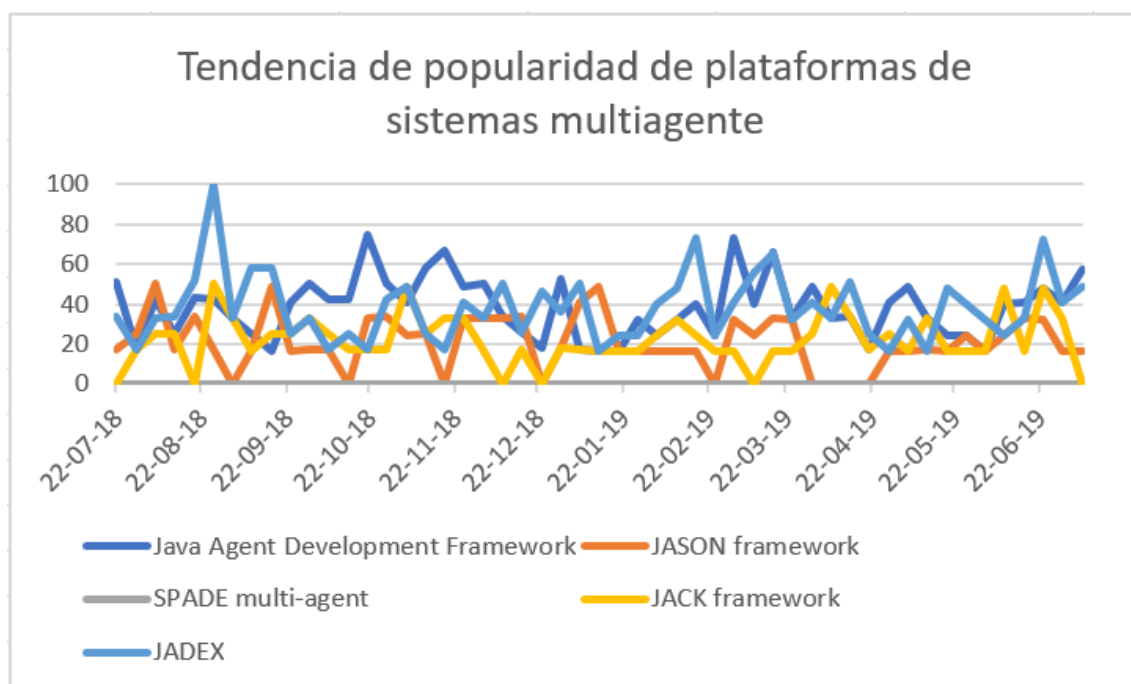


Figura 2.10: Tendencia de popularidad de las plataformas estudiadas basado en datos extraídos de Google Trends. El valor 100 refleja el término con mayor popularidad (Jadex) y el resto de datos muestran la popularidad en comparación con ese término, indicando 50 que el término es la mitad de popular, 0 que no es nada relevante, etc.

2.3.1. JADE

JADE ⁹ [9] (acrónimo de *Java Agent DEvelopment Framework*) es un framework para el desarrollo de agentes en Java que permite crear un sistema multiagente gracias a un middle-ware que sigue el estándar FIPA-ACL (acrónimo de *Foundation for Intelligent Physical Agents - Agent Communication Language*) [37, 36] para la comunicación de los agentes y el protocolo *FIPA-Agent Management* para la gestión de los agentes.

Para lograr su objetivo, JADE organiza los agentes en contenedores, que pueden encontrarse distribuidos en diferentes máquinas, y que son a su vez coordinados por un contenedor principal (ver Figura 2.11). Este framework dispone de una herramienta GUI para facilitar el trabajo del desarrollador (ver Figura 2.12).

En la actualidad JADE es uno de los frameworks más populares para el desarrollo de sistemas multiagente, por lo que encontrar recursos en la web es sencillo. Al estar desarrollado con Java se puede desplegar en cualquier sistema que disponga de la máquina virtual de Java (JVM). Esta plataforma ofrece una licencia LGPLv2 para su uso académico, por lo que podría ser empleada en el proyecto sin problema.

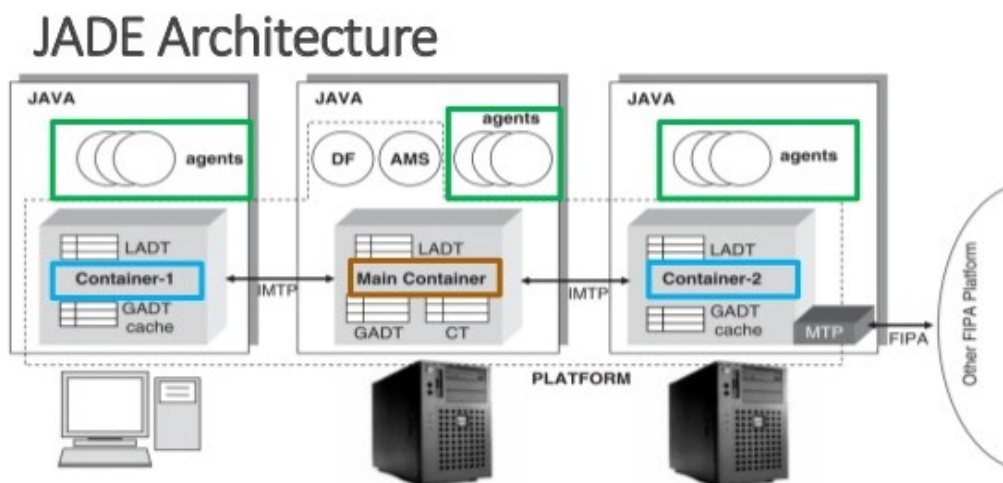


Figura 2.11: Arquitectura de JADE

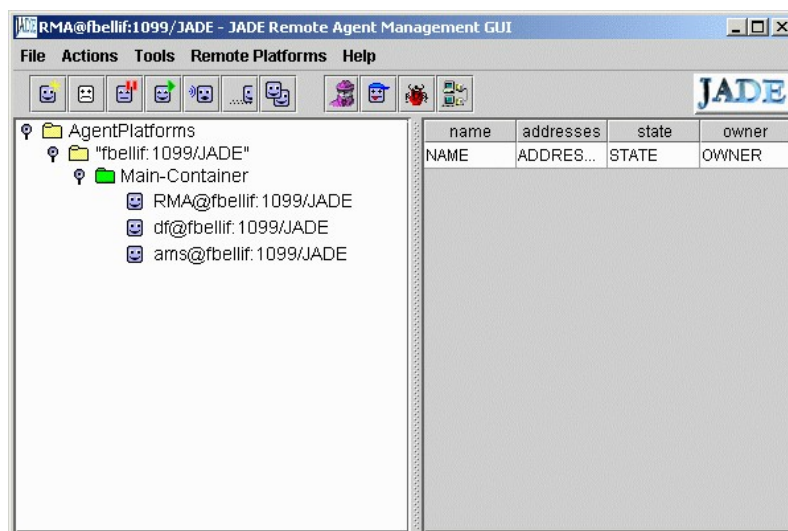


Figura 2.12: GUI de JADE

⁹<https://jade.tilab.com/>

2.3.2. Jadex

Jadex ¹⁰ [41] es la plataforma de agentes basada en el modelo BDI (Belief Desire Intention) [40] más popular en el mercado, llegando a ser tan reconocida y utilizada como JADE, con la que tiene una estrecha relación al tratarse de una extensión de la misma, de ahí su nombre. El objetivo inicial de Jadex era el de dotar a JADE de la capacidad de trabajar con agentes que siguiesen el modelo BDI y tratar de corregir algunos problemas encontrados en los protocolos FIPA [11].

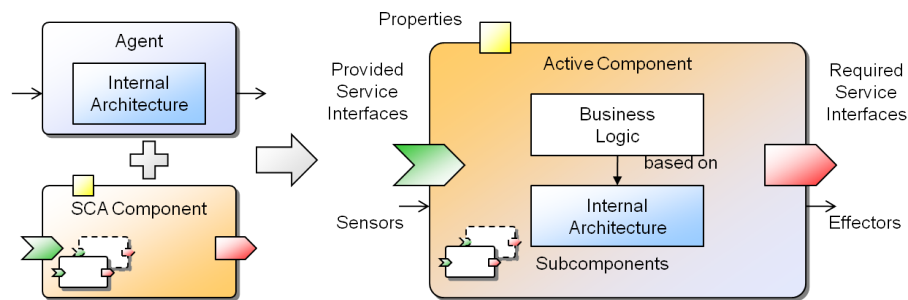


Figura 2.13: Arquitectura de Jadex

En la versión actual de Jadex, los agentes se basan conceptualmente en el modelo Service Component Architecture (SCA), una especificación para la creación de aplicaciones y sistemas utilizando una arquitectura orientada a servicios (SOA, del inglés Service Oriented Architecture), como se muestra en la figura 2.13. Esto permite que se puedan desarrollar agentes distribuidos, de manera concurrente, que interactúen entre ellos a través de servicios siguiendo el protocolo WSDL (Web Service Domain Language).

El lenguaje de desarrollo en Jadex es Java, utilizando XML para almacenar las características de los agentes, plataformas, etc. El distribuidor ofrece plugins para los IDE Eclipse y Android Studio, además de una aplicación independiente, llamada Jadex Control Center, que se puede utilizar para monitorizar el estado de nuestros agentes, como se observa en la figura 2.14.

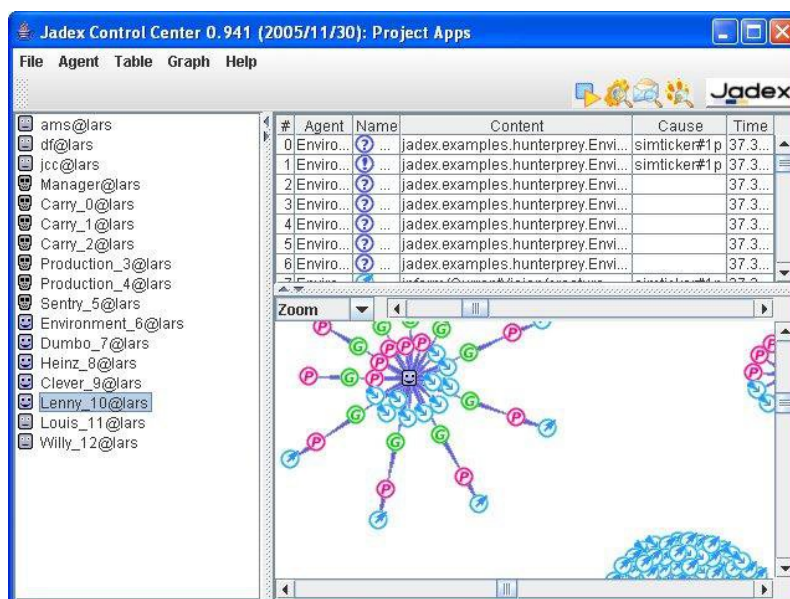


Figura 2.14: Jadex Control Center

¹⁰<https://github.com/actoron/Jadex>

Esta plataforma se ofrece bajo la licencia LGPLv3, por lo que podría ser utilizada en este proyecto sin problema y al realizarse la implementación en Java podría desplegarse en cualquier sistema que disponga de la JVM.

2.3.3. JACK

JACK ¹¹ [50] es la plataforma de carácter comercial más utilizada a la hora de trabajar con agentes basados en el modelo BDI. Aunque en los estudios utilizados como referencia se indica que JACK no sigue el protocolo FIPA, y tampoco se ha observado ninguna referencia en los manuales de usuarios que se encuentran en la web corporativa de Agent Software, se ha observado que sí aparece listado en la página web del propio protocolo ¹² y se ha encontrado la existencia de un plugin que extiende JACK para hacer que sus agentes sigan el estándar [52]. No se ha podido comprobar si en la última versión del producto se ofrece por defecto esa compatibilidad.

La programación en JACK se realiza en una modificación de Java denominada JACK Agent Language (JAL), que es una extensión de Java a la que se añaden las funcionalidades necesarias para facilitar la representación y desarrollo de los agentes. Para ello ofrecen un completo entorno de desarrollo llamado JACK Development Environment (ver figura 2.15) que permite, entre otras acciones, programar los agentes, diseñarlos de manera visual, monitorizar su estado, etc.

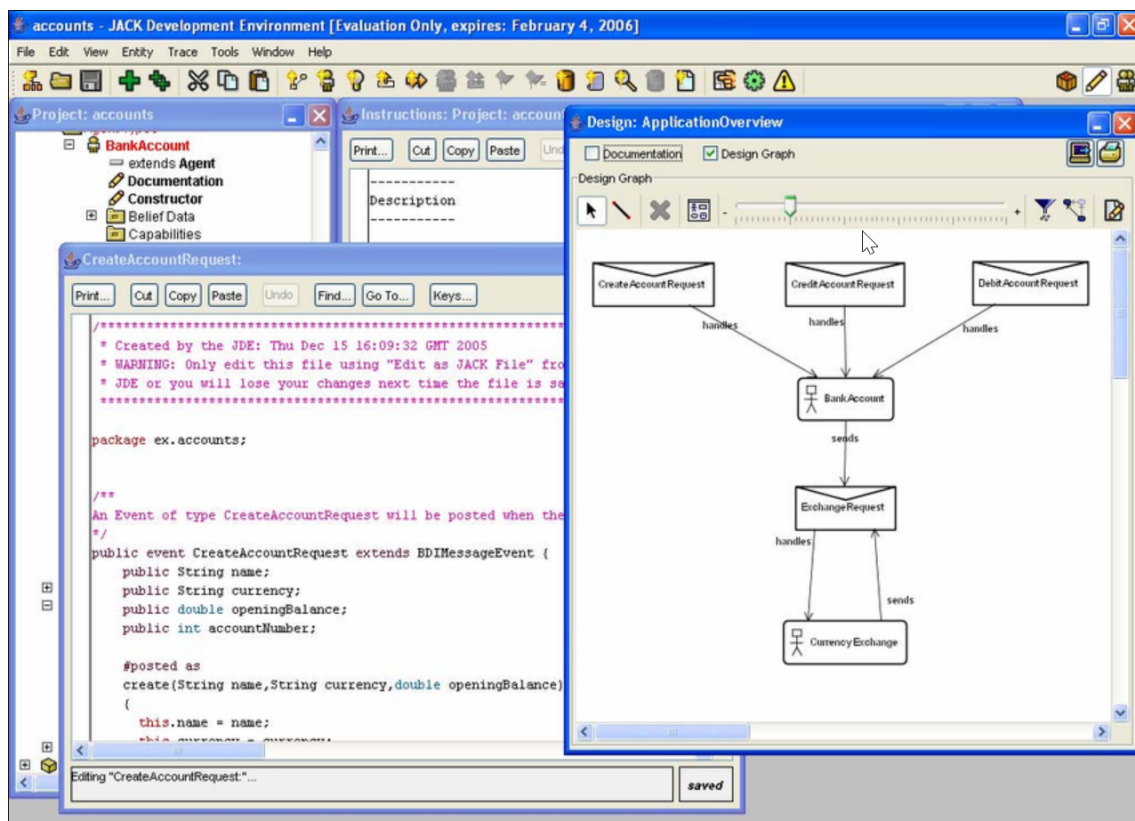


Figura 2.15: Jack IDE

El principal problema que muestra esta plataforma es que no ofrece una versión de código libre, aunque dispone de una versión de prueba temporal que hay que solicitar al distribuidor, por lo que en caso de querer utilizarla para este proyecto se debería adquirir

¹¹<http://www.agent-software.com.au/products/jack/>

¹²<http://fipa.org/resources/livesystems.html>

una licencia, algo que no sucede con el resto de herramientas analizadas que sí ofrecen versiones de libre uso.

2.3.4. JASON

JASON ¹³ (acrónimo creado a partir del nombre de su primera versión *Java-based Agentspeak interpreter used with Saci for multi-agent distribution Over the Net*) es un framework implementado en Java que, utilizando la semántica del lenguaje AgentSpeak, permite el desarrollo de sistemas multiagente que cumplen el estándar FIPA-ACL.

Por defecto JASON utiliza una infraestructura centralizada en la que todos los agentes se ejecutan en el mismo sistema, aunque opcionalmente permite al usuario utilizar JADE para que se encargue de la gestión y comunicación de los agentes en un sistema distribuido, obteniendo todos los beneficios que ofrece esa plataforma, como las herramientas de inspección de agentes.

JASON ofrece una modificación del IDE JEdit, que integra todas las herramientas necesarias para poder comenzar a trabajar con esta plataforma, como se muestra en la figura 2.16. Este IDE ofrece también una consola en la que se puede consultar el estado de los agentes y seguir paso a paso las instrucciones que está ejecutando, como se muestra en la figura 2.17.

JASON, al igual que Jadex y JACK, se basa en el modelo BDI. Al ser uno de los frameworks más conocidos y utilizados es sencillo encontrar multitud de recursos en la red. Al estar basado en Java se puede desplegar en cualquier plataforma que disponga de la JVM. JASON se ofrece bajo la licencia LGPLv3, por lo que podría ser utilizada en este proyecto.

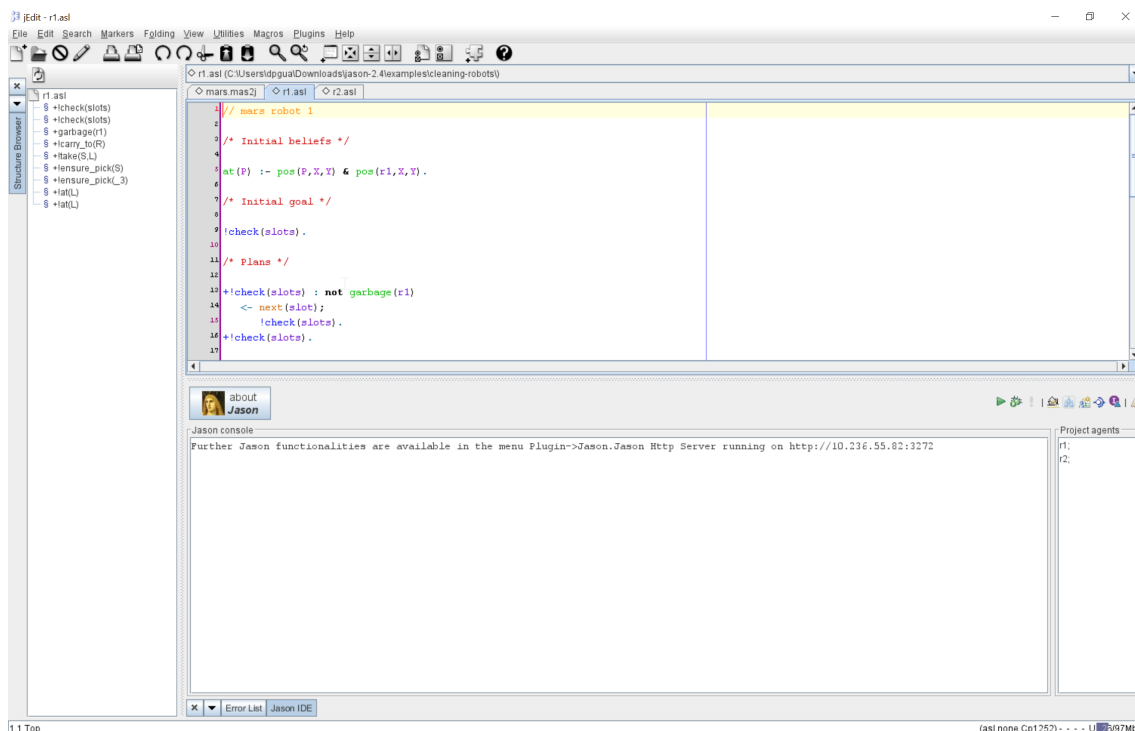


Figura 2.16: Modificación de JEdit para trabajar con JASON

¹³<http://jason.sourceforge.net/wp/>

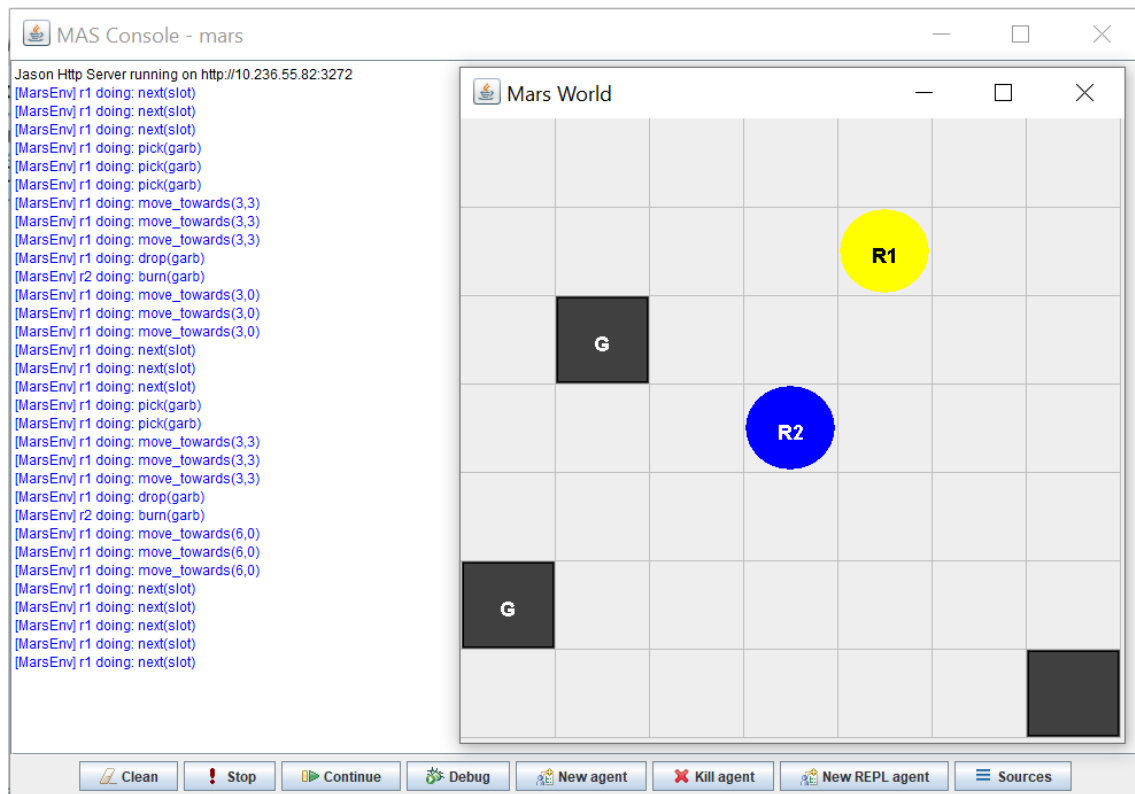


Figura 2.17: Consola de JASON. En la imagen se puede observar la interfaz gráfica de un programa en ejecución junto al detalle de las acciones que se han ejecutado

2.3.5. SPADE

SPADE¹⁴ (acrónimo de *Smart Python Agent Development Environment*) es un framework implementado en Python para el desarrollo de sistemas multiagente que cumplen el estándar FIPA. De igual manera que las tres anteriores alternativas, esta plataforma se basa en el modelo BDI para el diseño de los agentes. La principal diferencia con los otros frameworks analizados es que SPADE utiliza el protocolo XMPP para la comunicación de los agentes y para conseguirlo los planes utilizados se ofrecen a través de servicios a los que podrá acceder el agente para cumplir sus objetivos. Al igual que JADE, dispone de una interfaz web que permite consultar de manera sencilla el estado del sistema.

A pesar de no contar con la popularidad de las alternativas analizadas previamente, y por tanto resulta más complicado encontrar recursos en la red, el grupo de investigación GTI-IA tiene experiencia previa en su uso y dispone de todas las herramientas necesarias para conectar los agentes a la red social PESEDIA. Además este framework es desarrollado y mantenido por Javier Palanca Cámara¹⁵, miembro de la Universidad Politécnica de Valencia, por lo que en caso de encontrar problemas o tener dudas es posible comunicarse con él de primera mano, algo que no es posible con el resto de las plataformas.

Al estar desarrollado en Python puede ser ejecutado en cualquier sistema compatible y además ofrece vía web las herramientas necesarias para consultar el estado del sistema una vez esté en ejecución, como se muestra en la figura 2.18. Finalmente queda destacar que esta plataforma se ofrece bajo la licencia MIT, por lo que puede ser utilizado en nuestro proyecto sin problema.

¹⁴<https://spade-mas.readthedocs.io/en/latest/>

¹⁵<http://www.upv.es/ficha-personal/japaca1>

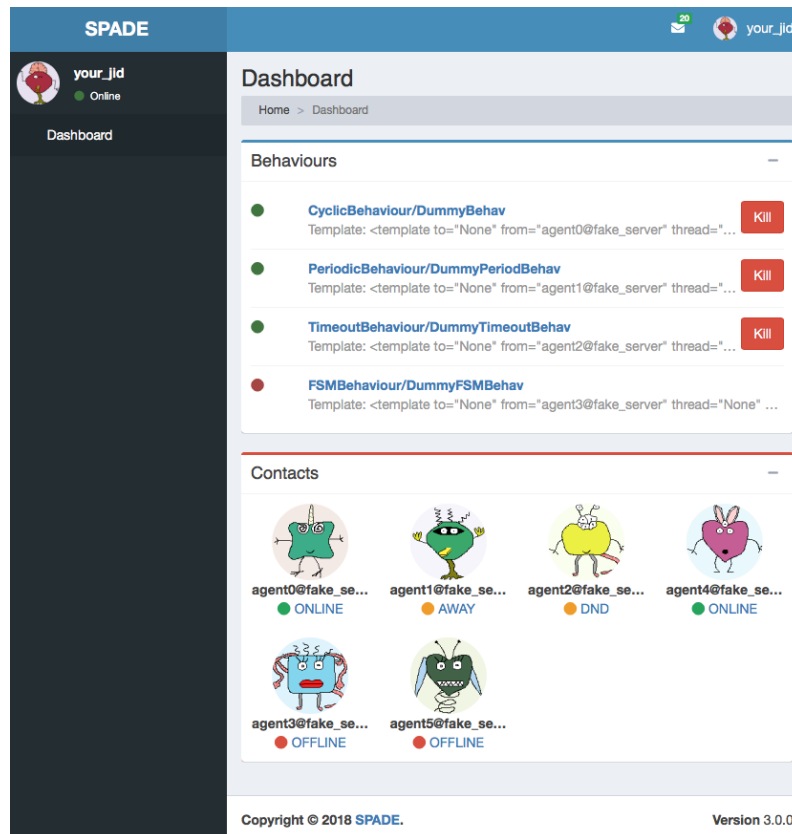


Figura 2.18: GUI de SPADE

2.3.6. Conclusiones

A pesar de ser la plataforma menos conocida de las analizadas, como ya se suponía en base a las propuestas seleccionadas en las que ni siquiera aparece mencionada y, adicionalmente, comprobado este aspecto utilizando la herramienta Google Trends, obteniendo los resultados que se muestran en la figura 2.19, se ha decidido seleccionar SPADE debido a que:

1. Al tener experiencia previa en el grupo de investigación con el uso de esta plataforma, se dispone de todos los componentes necesarios para su integración con PESEDIA.
2. El producto que se pretende generar estará basado en una arquitectura de micro-servicios web, muy apropiada para esta plataforma.
3. El desarrollador principal de SPADE trabaja en esta universidad y, por tanto, se puede entablar una comunicación de manera fluida con él en caso de encontrar problemas al utilizar la plataforma.

Todas estas ventajas hacen que las posibles características que se pueden encontrar en el resto de alternativas analizadas, mucho más populares, no sean suficientes para justificar su elección, aprendizaje y posterior adaptación e implementación en la red social. En la tabla 2.7 se puede consultar de manera sintetizada la información más relevante extraída en este punto.

	Lenguaje de programación	Dominio principal	Licencia	Popularidad
JADE	Java	Aplicaciones distribuidas compuestas por entidades autónomas	Comercial, Académica (LGPLv2)	Alta
Jadex	Java	Aplicaciones distribuidas compuestas por entidades autónomas	Comercial, LGPLv3	Alta
Jack	Java, JACK Agent Language (JAL)	Aplicaciones distribuidas compuestas por entidades autónomas	Comercial	Alta
JASON	Java	Aplicaciones distribuidas compuestas por entidades BDI autónomas	LGPLv3	Alta
SPADE	Python	Aplicaciones distribuidas basadas en mensajería instantánea usando el protocolo XMPP	MIT	Baja

Tabla 2.7: Comparativa de plataformas de sistemas multiagente

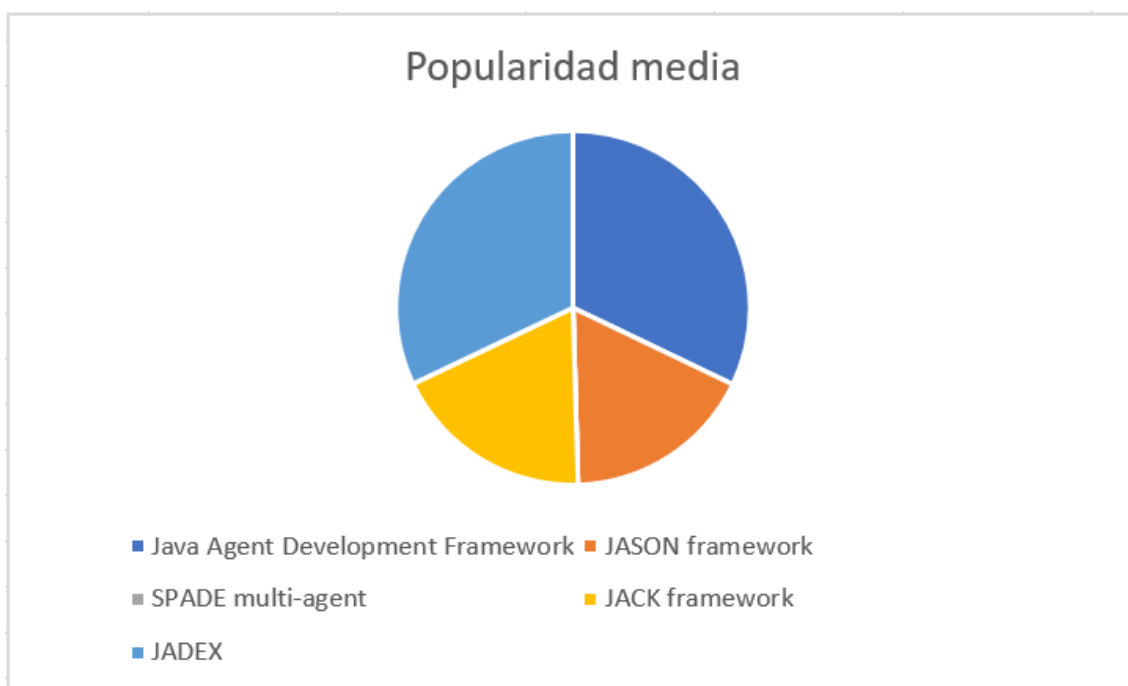


Figura 2.19: Comparativa de popularidad media de las herramientas analizadas basado en datos extraídos de Google Trends

2.4 Perfiles de usuario

Para poder proceder con el modelado y posterior desarrollo de los agentes es necesario extraer una serie de factores y características que definirán los diferentes tipos de usuarios que pueden coexistir en una red social. Una vez extraídos los factores se procederá a correlacionarlos entre sí.

El objetivo de este punto es tratar de extraer todos los factores que influyen en el comportamiento que el usuario tendrá dentro de la red social, tanto con acciones relativamente simples como podría ser publicar un comentario, dar un "me gusta" al conteni-

do de otros usuarios, actualizar su estado, etc., como con acciones complejas como serían añadir un amigo o unirse a un grupo según sus intereses, entre otras.

Los factores que hemos considerado, y que se detallan a continuación son:

1. Personalidad, edad y sexo
2. Privacidad y experiencia
3. Hábitos y patrones de uso en redes sociales

2.4.1. Personalidad, edad y sexo

Análisis

La personalidad [34], edad [39] y sexo [48] son los tres factores más importantes a la hora de determinar los patrones de uso que tiene un usuario en una red social y es necesario analizarlos en conjunto debido a la íntima relación que existe entre todos ellos [23]. Estos tres factores influyen claramente en las formas, expresiones y pautas verbales utilizadas al comunicarse [47], el número de amigos en los que estarán dispuestos a confiar y añadir [38], pasando por toda otra plétora de factores como la gestión de su perfil de privacidad [30] o las motivaciones para hacer uso de las redes sociales.

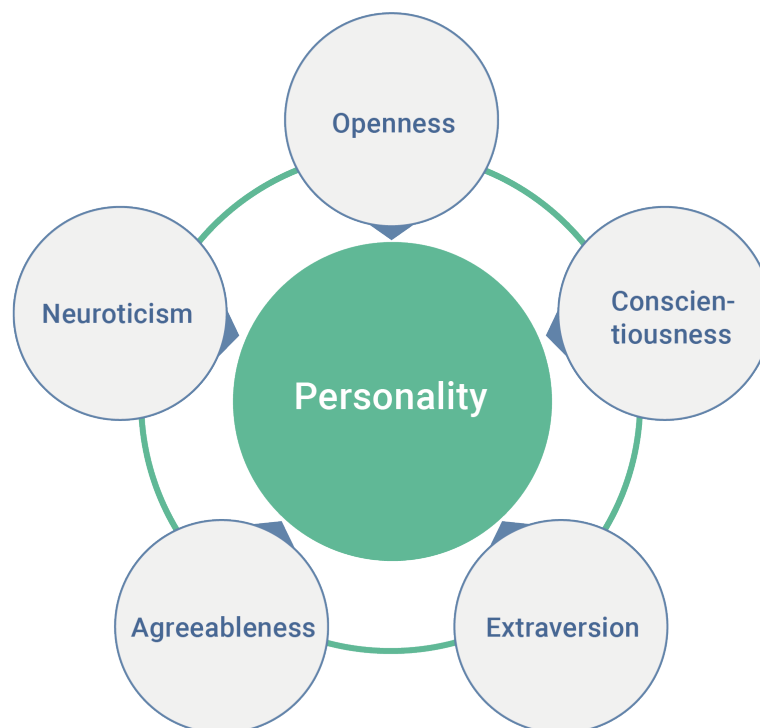


Figura 2.20: Modelo del Big Five

Existen multitud de modelos para representar perfiles de personalidad [53], siendo el Big Five [24] (modelo de los "cinco grandes", en castellano) uno de los más conocidos y utilizados a la hora de realizar estudios y análisis. Este modelo trata de definir la personalidad de la persona estudiada en torno a cinco grandes rasgos independientes y, a la vez, complementarios (ver figura 2.20). A continuación se detallan estos rasgos y su influencia en el uso de las redes sociales:

- **Apertura al cambio (*Openness to experience*):** este factor determina la disposición de una persona a enfrentarse a opiniones que contradigan su forma de ver y entender el mundo y adquirir nuevos conocimientos o vivir experiencias saliendo de su zona de confort. Los usuarios de redes sociales que tienen un valor alto en este factor tienden a estar dispuestos a probar nuevos métodos de comunicación, por lo que pueden llegar a hacer un uso intenso de las redes sociales e incluso a utilizar varias de manera simultánea [44]. Tienen tendencia a hacer muchos amigos con rasgos tanto de apertura al cambio como de extraversión [33], lo que provoca que acaben teniendo relaciones sociales dispersas. Este tipo de usuarios tiene una actitud laxa respecto a la configuración de sus perfiles de personalidad y por tanto muestra una tendencia a revelar más información privada [4].
- **Responsabilidad (*Conscientiousness*):** este factor representa la capacidad que tiene una persona para ser organizada y disciplinada. Los usuarios con este rasgo tienden a hacer un uso menos intenso y regular de las redes sociales [49] [45] y tienen más cuidado con los contenidos que generan y la cantidad de información personal que exponen [34].
- **Extraversión (*Extraversion*):** la extraversión es el rasgo de personalidad que dictamina la tendencia de una persona a relacionarse con otras. Los usuarios con un factor elevado de extraversión tienen tendencia a hacer un uso más intenso y prolongado de las redes sociales [49], tienden a tener un número mayor de amigos en la plataforma [44] y a pertenecer a un mayor número de grupos [26]. Además, los usuarios extravertidos utilizan las redes sociales como una herramienta para mantenerse en contacto con sus conocidos, aunque no las utilizan como un reemplazo para las interacciones reales [5], por lo que tienden a compartir con ellos una mayor cantidad de información personal tanto a través de los "muros" como de manera directa a través de sistemas de comunicación instantánea tipo chat. También suelen subir más fotos [15], pero paradójicamente comparten menos información personal de manera pública [4]. Finalmente, y como ya se ha visto, este factor influye en gran medida en el comportamiento de un usuario en la red social, pero no obstante no existen datos estadísticos significativos que demuestren su influencia en la gestión de sus perfiles de privacidad [12].
- **Cordialidad (*Agreeableness*):** este factor determina la capacidad del individuo para alcanzar acuerdos amistosos y mantener relaciones interpersonales positivas con otras personas. Como consecuencia de este rasgo los usuarios tienden a realizar más publicaciones sobre sí mismos y suben gran cantidad de fotos en las que suelen salir acompañados de más personas [4]. Tienden a comunicarse con personas abiertas al cambio, extravertidas o que sean cordiales como ellos [33].
- **Neuroticismo (*Neuroticism*):** este factor representa la tendencia de una persona a tener sentimientos negativos, siendo este rasgo popularmente conocido como inestabilidad emocional. Las personas con rasgos neuróticos tienden a pasar más tiempo conectadas a las redes sociales y compartiendo información personal con la intención de resultar interesantes ante los demás [29]. Las personas con este rasgo social tienden a hacer uso de chats [3] y otro tipo de herramientas de comunicación instantánea [21] y, dentro de las redes sociales, su apartado favorito es habitualmente los "muros" [44]. Este grupo de usuarios suele escribir sobre temas personales pero suele compartir menos fotos fuera de su perfil [4].

Existen multitud de estudios alrededor del Big Five y los posibles perfiles sociales más habituales que pueden surgir en torno a la agrupación y notoriedad de los distintos factores previamente explicados. En el trabajo de Gerlach et al. "A robust data-driven

approach identifies four personality types across four large data sets. Nature Human Behaviour” [23] se encuentra uno de los modelos propuestos más recientes. En dicho trabajo los autores detallan cuatro grandes perfiles de personalidad (ver figuras 2.21 y 2.22) que se exponen de manera sintetizada a continuación:

- *Self-centered* (Egocéntrico): las personas clasificadas como egocéntricas tienden a ser extravertidas, menos conservadoras y más conflictivas e irresponsables. Este grupo está compuesto principalmente por hombres y mujeres jóvenes (mayoritariamente por hombres) y tienden a cambiar a otros perfiles una vez entrada en la etapa de madurez, motivo principal por el que la mayoría de mujeres adoptan otro perfil de personalidad antes que los hombres.
- *Role model* (Modelo a seguir): este grupo posee los factores de personalidad más deseables, de ahí su nombre. Las personas que poseen este perfil poseen altos valores de extraversión, cordialidad y responsabilidad y un menor nivel de neuroticismo. Es el perfil adoptado en el proceso de maduración y con el crecimiento constante más notable conforme avanza la edad, comenzando a mitad de la veintena y extendiéndose hasta la vejez. Este perfil es notablemente más presente en mujeres que en hombres.
- *Reserved* (Reservado): este perfil muestra unos niveles balanceados en la mayoría de factores a excepción del neuroticismo y la apertura al cambio, que son inferiores a los niveles vistos en los otros perfiles. Este grupo lo componen principalmente hombres y mujeres de mediana edad, teniendo menor representación que el resto de perfiles.
- *Average* (Promedio): este perfil sirve para aglutinar y representar a la mayoría de personas que no tienen cabida en el resto de perfiles, ya que describe a alguien bastante cordial y responsable, algo extravertido y neurótico pero también un poco conservador.

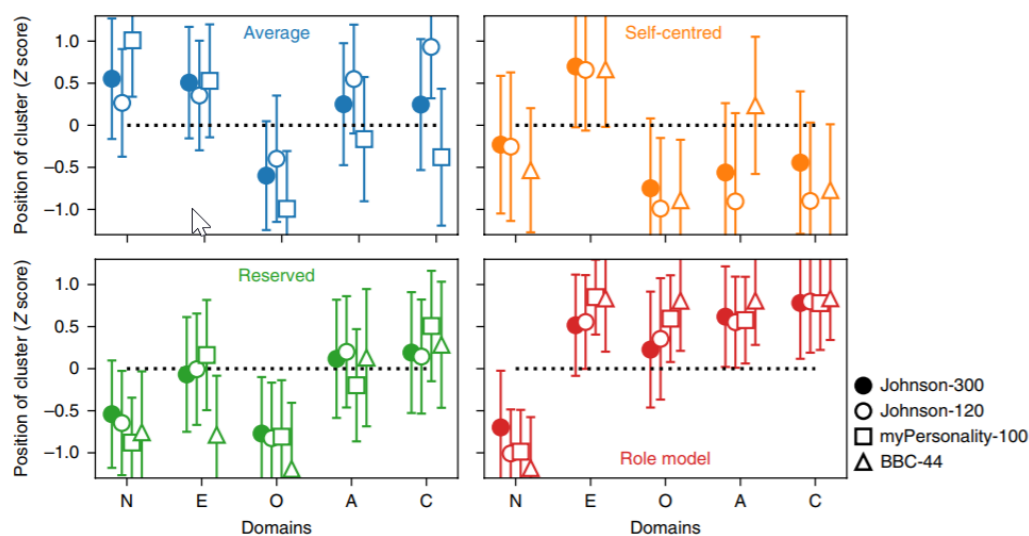


Figura 2.21: Clustering de los datasets empleados por Gerlach et al. para extraer y proponer sus perfiles de personalidad en base a los factores del Big Five. Los símbolos utilizados sirven para diferenciar los cuatro datasets empleados en el estudio. Fuente [23]

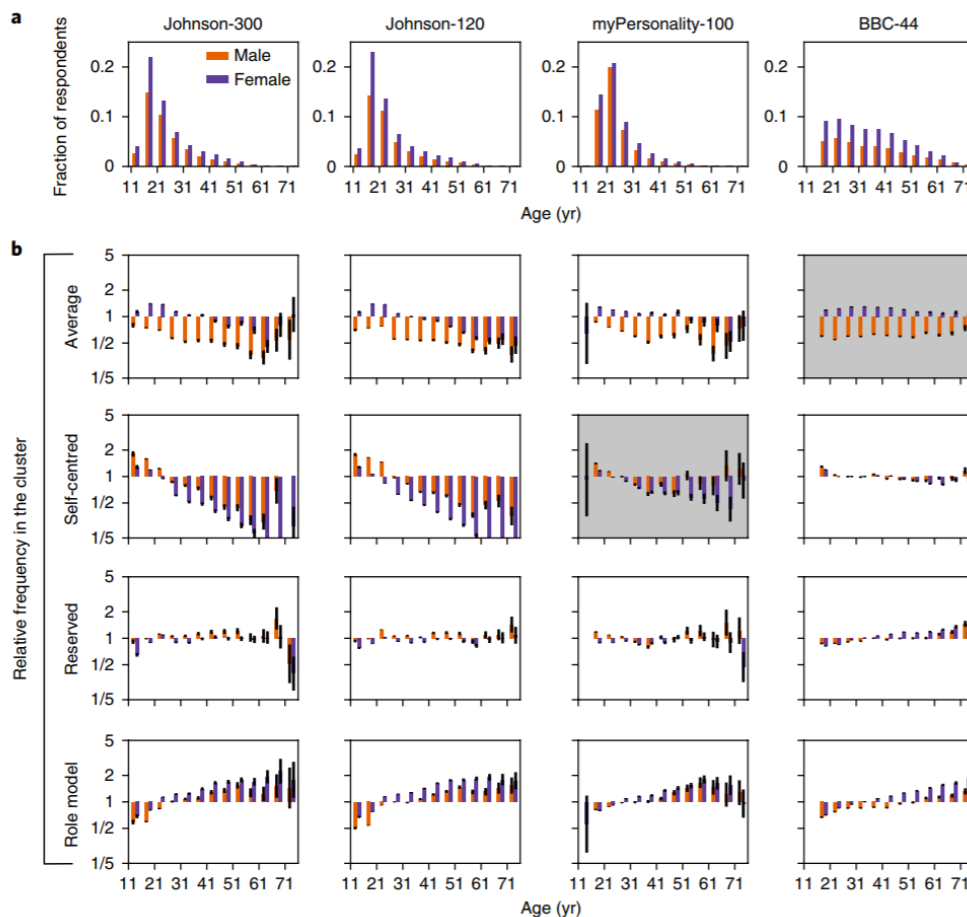


Figura 2.22: Correlación entre sexo, edad y perfiles propuestos por Gerlach et al. En la imagen A se puede observar la distribución por edades y sexos en los diferentes datasets. En la figura B se observa su frecuencia dentro de los cuatro perfiles, siendo las muestras poco representativas indicadas con fondo en gris. Fuente [23]

Conclusiones

Una vez completado el análisis de estos tres factores clave y analizadas las propuestas más relevantes - y también vanguardistas - en la clasificación de personas según su perfil de personalidad, se ha decidido utilizar el modelo propuesto en el estudio de Gerlach et al. [23] para etiquetar a nuestros agentes en base a las razones que se exponen a continuación:

1. Las fuentes de datos utilizadas: en la mayoría de estudios de este tipo se han encontrado muestras de datos que pueden resultar poco representativas, mientras que en la propuesta de Gerlach et al. [23] se usan cuatro datasets: Johnson-300, Johnson-120, myPersonality-100 y BBC-44 (todos detallados y accesibles para su descarga en el artículo), que tienen muestras lo suficientemente pobladas como para ofrecer información y conclusiones representativas, siendo la muestra más pequeña de 145.388 datos (el dataset Johnson-300) y la más grande de 575.380, correspondiente al dataset myPersonality-100, con el beneficio añadido de estar poblados por datos generados a partir de una demografía amplia, compuesta por personas de diferentes edades e incluso países.
2. Consistencia con otros estudios previos: los estudios más conocidos en cuanto a agrupación de perfiles de personalidad plantean tres grandes tipos conocidos co-

mo ARC por los nombres de sus autores (Asendorpf [7], Robins [43] y Caspi [13]), a saber: *resilient*, *overcontrolled* y *undercontrolled*, utilizados habitualmente en el estudio de niños, pero también aplicable en adultos y muy discutidos por la comunidad científica [20] [16] debido en gran parte a la dificultad para replicar los experimentos y a la inconsistencia de los resultados. Los cuatro modelos resultantes de la propuesta de Gerlach et al. [23] finalmente parecen apoyar y ampliar el modelo ARC, pero utilizando una muestra lo suficientemente significativa como para poder replicarla y comparar los resultados, principales puntos de crítica del modelo ARC.

3. **Facilidad para ser modelados:** al ya estar correlacionada la información por sexo, edad y perfil emocional de los usuarios, el modelo propuesto por Gerlach et al. [23] se plantea como una buena alternativa a la asignación arbitraria de valores basados únicamente en el Big Five, realizando la asignación manual de probabilidades de mayor o menor frecuencia de cada rasgo en base a los sexos y edades que se busca simular, obteniendo así modelos que se aproximen más a la personalidad de los usuarios reales que se pretende representar.

2.4.2. Privacidad y experiencia

Análisis

Para decidir qué acciones podrán realizar los agentes y la probabilidad con las que decidan hacerlas se necesita más información aparte de la edad, el sexo y el perfil de personalidad a modelar. A continuación se analizan otros de los principales factores clave a la hora de determinar los patrones y hábitos de uso en una red social, como serán el celo que tenga el usuario en cuanto al alcance de su información y publicaciones así como la gestión de las opciones de configuración relativas a la privacidad que se ofrecen en las mismas y finalmente su experiencia en el uso de redes sociales.

La privacidad es uno de los factores más determinantes para los usuarios de una red social [31] y es evidente que existe una gran relación entre la concepción que tiene el usuario en cuanto a la importancia de sus datos, la difusión que le gustaría hacer de los mismos y el uso de las opciones de configuración que le facilitan cumplir sus objetivos [27]. A pesar de que las diferentes redes sociales ofrecen cada vez más opciones para que el usuario pueda definir su concepción de privacidad, es muy frecuente encontrar usuarios que sienten una gran preocupación por su privacidad pero no hacen uso (o hacen un uso incorrecto) de estas herramientas. Este fenómeno es conocido como la "paradoja de la privacidad" [8, 35]. Existen multitud de motivos que justifican esta paradoja, como puede ser la sensación de saturación que tiene el usuario al ver la gran variedad de opciones que se le ofrece [31], la facilidad que puede tener para entender la relevancia que tiene su información privada o, simplemente, su motivación para configurar las opciones disponibles [25]. Existen multitud de estudios que tratan de encontrar soluciones a esta problemática, que quedan fuera del objetivo de este trabajo.

De entre la multitud de estudios disponibles que tratan de generar perfiles de usuario según la correlación entre su preocupación por la privacidad y sus acciones para tratar de gestionarla se ha decidido destacar dos, ya que en ambos se propone y detalla una clasificación de perfiles empleando "etiquetas" que servirán para definir a los agentes. A continuación se explica qué modelos propone cada estudio.

En el primero de los estudios, Schomakers et al. [46] determinan tres perfiles de usuario a partir de sus acciones para preservar su privacidad y la importancia que tiene para ellos, como se puede observar en la figura 2.23:

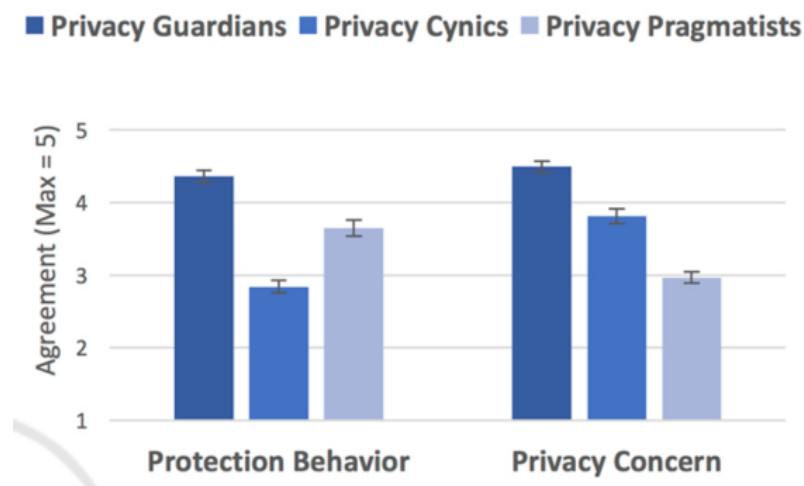


Figura 2.23: Relación entre acciones y preocupación para preservar la privacidad. Fuente [46]

- *Privacy guardians* ("Guardianes"): este grupo es el que demuestra una mayor preocupación por la privacidad de sus datos y también el que más tiempo invierte en intentar garantizarla, siendo los que menos confían en que las empresas gestionen sus datos de manera correcta, en gran parte por haber tenido experiencias negativas en el pasado. Las principales razones que argumentan para no gestionar sus opciones de privacidad son la complejidad y el sentimiento de ineficacia de las mismas. Además, son el grupo que menos percepción tiene de recibir un beneficio por parte de las empresas a cambio de su información personal.
- *Privacy cynics* (Cínicos): este segundo grupo se caracteriza por su alta preocupación en todo lo referente a la privacidad y su escasa actuación para tratar de asegurarla, siendo claramente el grupo que mejor refleja la "paradoja de la privacidad". Destaca por ser el grupo que toma las peores y menos eficaces decisiones en lo relativo a la privacidad de sus datos. Sus principales motivos para no configurar las opciones de privacidad correctamente son la complejidad y el tiempo que puede llegar a consumir en contraposición a los resultados que van a obtener.
- *Privacy pragmatists* (Pragmáticos): el tercer y último grupo es el que menor preocupación tiene por todos los aspectos relativos a la privacidad, pero que aún así realiza acciones para intentar garantizarla. Se caracterizan por ser el grupo que menos cree necesitar tener privacidad en Internet, lo que puede deberse a que son los que menos malas experiencias han sufrido. Destacan por ser el grupo que realiza las acciones más eficaces para preservar su privacidad. Es el grupo que más confía en que las empresas garanticen la privacidad de sus datos y los que más satisfechos se sienten por los beneficios que obtienen tras ceder su información personal.

El segundo estudio, propuesto por Wisniewski et al. [51], agrupa a los usuarios en seis perfiles según sus estrategias para gestionar las opciones de privacidad y otros seis perfiles según su experiencia gestionando las opciones de privacidad, con el objetivo de encontrar una relación entre ambas. A continuación se muestran los seis perfiles según la estrategia utilizada, ordenados según el porcentaje de usuarios representados (ver figura 2.24):

- *Privacy balancers* (Equilibrados): este perfil es el que cuenta con mayor representación, correspondiendo al 36 % de la muestra empleada. Los usuarios equilibrados tratan de utilizar la mayoría de opciones disponibles en la plataforma, en menor

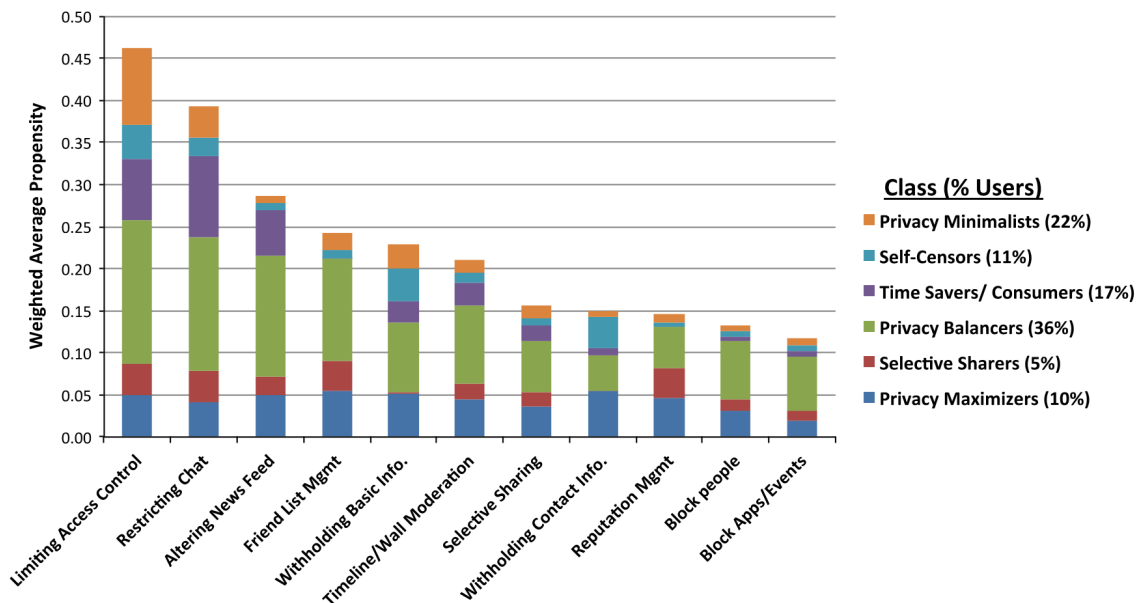


Figura 2.24: Relación entre perfiles de privacidad y acciones realizadas para preservarla. Fuente [51]

medida que los usuarios de otros grupos que se detallan a continuación, tratando de alcanzar un compromiso entre el resultado de las acciones realizadas y tiempo invertido, de ahí el nombre del grupo.

- *Privacy minimalists* (Minimalistas): los minimalistas representan al 22 % de los encuestados. Es el grupo que menos acciones toma para tratar de garantizar su privacidad, limitándose a las más básicas y comunes. Sus acciones primarias para tratar de garantizar su privacidad son limitar el alcance de su información y restringir quién puede comunicarse con ellos a través de chat, quedando en lugar secundario la cantidad de información pública que muestran en su perfil y la gestión de su muro y amigos, tendiendo a no bloquear invitaciones a eventos y solicitudes de amistad de otras personas.
- *Time savers/Consumers* (Optimizadores): el perfil del optimizador representa al 17 % de la muestra. Este perfil de usuario es casi idéntico al de los minimalistas pero cambia el objetivo por el que deciden hacer uso de las redes sociales. Para este grupo de usuarios las redes sociales son una herramienta de consumo y acceso a la información, por lo que hacen especial énfasis en impedir que otros usuarios puedan molestarles mientras hacen uso de las mismas, así como de garantizarse que la información que les sugiere el sitio es de su interés.
- *Self-censors* (Auto censores): este grupo representa al 11 % de los encuestados. Los usuarios auto censores actúan de una manera muy similar a los minimalistas y tienden a utilizar únicamente las opciones justas y necesarias para garantizar su privacidad, prefiriendo no publicar contenido sensible en lugar de limitar el alcance del mismo y tratan de cuidar al máximo el contenido de sus publicaciones [17]. Las acciones principales que utilizan para garantizar los miembros de este grupo, aparte de la ya mencionada auto censura y las acciones realizadas por los usuarios de perfil minimalista, con los que tienen mucho en común como ya se ha mencionado, es limitar el alcance de su información personal y de contacto, o directamente no publicarla.

- *Privacy maximizers* (Maximizadores): los maximizadores representan únicamente al 10% de los usuarios. Estos usuarios son los que optan por hacer un uso más intenso de las opciones de privacidad y juntan los patrones de comportamiento vistos en los usuarios de los grupos analizados previamente sin importarles el tiempo que tengan que invertir en realizar las acciones que garanticen su privacidad.
- *Selective sharers* (Distribuidor selectivo): el perfil de distribuidor selectivo es el menos común según los resultados mostrados en la propuesta, representando únicamente al 5% de los usuarios. Los usuarios de este grupo optan por elegir cuidadosamente quien tiene acceso a su información y publicaciones, por lo que tienden a preocuparse menos por la cantidad de información personal que publican o por el contenido de sus mensajes, ya que siempre tratan de dirigirlos a las personas que consideran correctas, ya sea limitando la cantidad de amigos que tienen como creando listas de amigos personalizadas.

Aparte de los perfiles según las acciones de privacidad empleados por el usuario, la propuesta de Wisniewski et al. [51] también aporta una serie de perfiles según su experiencia, conocimiento y manejo de estas opciones, como ya se ha comentado. La diferencia fundamental en la categorización de cada perfil es la cantidad de opciones que conocen aunque no hagan uso de ellas, siendo incremental entre un grupo y el siguiente con más experiencia. A continuación se detallan estos perfiles, que también se pueden consultar en la figura 2.25 ordenados de menor a mayor experiencia, junto a la relación con los perfiles de privacidad que se acaba de analizar y mostrado en la figura 2.26:

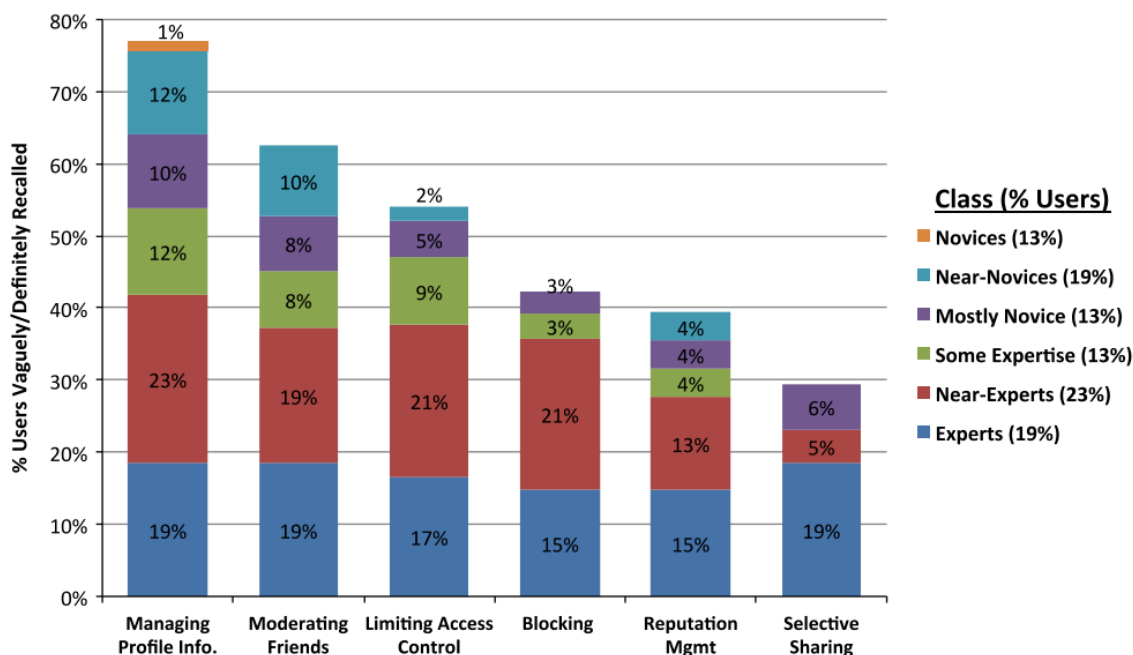


Figura 2.25: Relación entre perfiles de experiencia gestionando opciones de privacidad y acciones/opciones reconocidas por el usuario para preservarla. Fuente [51]

- *Novices* (Novatos): son los usuarios que menos opciones de privacidad conocen y es uno de los grupo con menor representación, el 13% de los encuestados. Está compuesto en mayor medida por usuarios de tipo equilibrado y auto censores.
- *Near-novices* (Casi novatos): es el segundo grupo con mayor representación, con el mismo porcentaje (19%) que el perfil experto. En este grupo se observa una clara

mayoría de usuarios con perfiles de privacidad de tipo optimizador y minimalista y, en menor medida, de tipo auto censor.

- *Mostly novice* (Algo novatos): segundo de los grupos con un 13% de representación. Compuesto principalmente por usuarios de tipo minimalista y autocensor y, en menor medida, por los del tipo optimizador.
- *Some expertise* (Algo expertos): tercer y último grupo con un 13% de representación. Es el grupo más equilibrado de todos en cuanto a distribución de perfiles de privacidad, aunque una vez más prevalecen los perfiles de tipo equilibrado y en menor medida minimalistas y optimizadores.
- *Near-experts* (Casi expertos): el grupo de los casi expertos es el que más representación tiene siendo el 23% de los encuestados miembros del mismo. Se observa una clara mayoría de usuarios de tipo balanceado, aunque también hay en menor medida maximizadores y distribuidores selectivos.
- *Experts* (Expertos): este es el grupo de usuarios que mayor cantidad de opciones de privacidad conocen y está compuesto mayoritariamente, al igual que el de casi expertos, por los usuarios con perfiles tipo equilibrado y, en menor medida, por maximizadores y distribuidores selectivos. Es el segundo grupo en representación siendo el 19% del total de los encuestados, igual que el de casi novatos.

	Privacy Maximizers	Selective Sharers	Privacy Balancers	Time Savers	Self-Censors	Privacy Minimalists
Experts	13 (5.6) +	6 (3) +	28 (20.4) +	4 (9.4) -	1 (6.1) -	5 (12.6) -
Near-Experts	11 (6.9) +	8 (3.7) +	31 (25.4) +	7 (11.8) -	4 (7.6) -	10 (15.7) -
Some Expertise	1 (4) -	0 (2.1) -	17 (14.6) +	9 (6.8) +	4 (4.4) -	10 (9.1) +
Mostly Novice	1 (3.8) -	0 (2) -	1 (13.9) -	4 (6.5) -	12 (4.2) +	21 (8.6) +
Near-Novices	1 (5.8) -	0 (3.1) -	1 (21.4) -	22 (9.9) +	7 (6.4) +	19 (13.2) +
Novices	3 (3.9) -	2 (2.1) -	22 (14.3) +	5 (6.6) -	5 (4.3) +	3 (8.8) -

Figura 2.26: Relación entre perfiles de privacidad y niveles de experiencia de los usuarios. Entre paréntesis se indica la diferencia respecto a la media en la relación entre grupos, también apoyado por el color de la celda. Fuente [51]

Como se puede observar, los usuarios con perfiles tipo equilibrado son los que más representación muestran en la mayoría de grupos según su experiencia, ya que es el perfil con mayor población, casi 4 de cada 10 encuestados. Como queda claro que no hay una relación directa entre perfiles, a la hora de diseñar los modelos se tendrán en cuenta todas las posibles combinaciones que puedan surgir utilizando como referencia otros factores, como el sexo y el grupo de edad del usuario o sus patrones de uso, que se verán más adelante.

Conclusiones

Para modelar nuestros agentes se ha decidido utilizar los modelos planteados en Wisniewski et al. [51] por las siguientes razones:

1. En primer lugar por la flexibilidad que ofrece al proporcionar de manera desacoplada la información relativa a la experiencia del usuario y las estrategias utilizadas al gestionar sus perfiles de privacidad. Mientras que en el primer trabajo (Schomakers et al. [46]) los modelos propuestos integran ambos análisis, el trabajo seleccionado los ofrece por separado, lo que permite generar más combinaciones en los modelos que se propondrán, sabiendo que se siguen respetando los resultados originales, mientras que si se utilizasen los modelos propuestos por Schomakers et al. [46] se perdería esa posibilidad.
2. El trabajo de Wisniewski et al. [51] muestra una serie de acciones que realizarán los usuarios según su perfil, mientras que en el de Schomakers et al. [46] solamente se encuentran las razones por las que se realiza la categorización de los usuarios. Esta información es clave ya que, aunque posteriormente se tendrán que sopesar y definir qué acciones son válidas en PESEDIA - una red social diferente a la utilizada en la realización del estudio - proporciona las herramientas para poder definir probabilidades según los perfiles estudiados.
3. El trabajo de Wisniewski et al. [51] es más relevante y se ha usado en más estudios como referencia y, al tener ambos estudios un número de muestras similar (337 en el caso del primer estudio y 314 para el segundo), se ha decidido considerar como factor clave la relevancia de los mismos, a diferencia de la selección de modelos de personalidad visto previamente en la que se ha optado por elegir el estudio con un muestreo mayor.

2.4.3. Hábitos y patrones de uso en redes sociales

Análisis

Al igual que en los puntos anteriores, este es un campo estudiado en multitud de ocasiones y sin acuerdos claros entre los diferentes autores. En este punto se analizarán cuatro de las propuestas más relevantes en cuanto a categorización de usuarios según sus hábitos y patrones de uso. Los trabajos se mostrarán en orden, de más antiguo a más reciente, con el objetivo de ver si existen cambios en las pautas de uso con el paso de los años y si varían según sobre qué red social se esté usando como referencia.

La primera propuesta corresponde a los autores Pfeil et al. [39]. En este trabajo los autores tratan de analizar y encontrar patrones de uso basándose en la edad del usuario y utilizando como referencia la antigua red social MySpace. A continuación se detallan algunas de las conclusiones extraídas:

- Los jóvenes tienen grupos de amigos más amplios y con edades aproximadas a las suyas, mientras que los adultos tienden a tener amigos en grupos de edades más variadas incluyendo mayoría de gente joven, algo que puede deberse únicamente al ámbito de la red social (enfocada para jóvenes) y al momento en el que se realizó el estudio.
- Las mujeres, en todas las edades, tienen de media más amigos que los hombres y sus amigos son habitualmente más hombres que mujeres, mientras que en el caso de los hombres el número de amigos se reparte equitativamente entre los dos sexos.

- Las adolescentes crean de media más entradas tipo blog que su contraparte masculina. Esta diferencia es menos notable en adultos.
- Los varones jóvenes comparten más contenido de tipo audiovisual que las mujeres, mientras que en el grupo de edad adulto son las mujeres las que comparten más contenidos de este tipo.
- Las mujeres reciben más comentarios que los hombres en todas las franjas de edad.

En el trabajo de Correa et al. [15] los autores tratan de encontrar una relación entre los factores de la personalidad (empleando el Big Five) y las pautas de uso de una red social. Aunque ya se ha hecho alguna referencia previa en este documento, a continuación se detallan las conclusiones principales extraídas de este trabajo:

- Como norma general las personas con valores altos de extraversión, neuroticismo y apertura al cambio hacen más uso de las redes sociales, siendo la extraversión el elemento con mayor influencia.
- En el caso específico de los hombres se cumple lo visto en el anterior punto, pero el factor de apertura al cambio pasa a ser el menos relevante a la hora de comprobar si usará o no una red social. Las mujeres, en cambio, tienen como factor menos determinante el neuroticismo a la hora de determinar su uso de la red social.
- Entre los jóvenes (18-29 años) la extraversión es el factor clave a la hora de influir de manera significativa en el uso de las redes sociales.
- En el caso de los adultos (≥ 30 años) se cumple perfectamente lo visto en el primer punto.

El tercer trabajo a analizar, de Constantinides et al. [14] trata de obtener los perfiles de usuario de redes sociales en los Países Bajos con el objetivo de generar etiquetas para categorizar a los usuarios:

- *Beginner user* (Usuario principiante): este primer perfil también es el que representa a la mayoría de población, casi el 50 % de los entrevistados. Su principal motivación al utilizar las redes sociales es para mantener contacto con conocidos y familiares y no generan mucho contenido en las mismas. Sus principales actividades son enviar mensajes privados a sus amigos, buscar gente conocida, compartir fotos y actualizaciones de perfil. El perfil tipo de este grupo de usuarios es una mujer entre los 25 y los 34 años.
- *Habitual user* (Usuario habitual): el segundo grupo representa al 18.2 % de los usuarios. La principal diferencia con los usuarios principiantes en cuanto a motivación para utilizar las redes sociales radica en que los usuarios habituales ven las redes sociales, además de como una herramienta para socializar, como una herramienta para compartir y recibir información, lo que influye en sus patrones de uso al ser mucho más activos. El perfil tipo de este grupo de usuarios es un hombre de mediana edad, entre los 25 y los 44 años.
- *Outstanding user* (Usuario sobresaliente): los usuarios sobresalientes (26.2 % del total) tienen unos hábitos casi idénticos a los de los usuarios habituales, siendo las principales diferencias entre los dos grupos el número de amigos que tienen, superior al del grupo anterior y la eliminación del uso de las redes sociales como medio de información. El perfil tipo de este grupo es el mismo que el de los usuarios principiantes.

- *Expert user* (Usuario experto): es el grupo con menor representación de todos, únicamente el 10.5 % de la muestra. Sus motivaciones y acciones en la red sociales son muy similares a las de los usuarios habituales radicando la principal diferencia en el tiempo que le dedican diariamente a consumir redes sociales, siendo el grupo más activo de los cuatro también a la hora de generar contenido. El perfil tipo de este grupo de usuarios son mujeres jóvenes entre los 16 y 24 años.

Finalmente, la última propuesta que se analiza, planteada por Alarcón et al. [1], trata de continuar los análisis vistos en el anterior trabajo sin centrarse en una zona geográfica concreta, por lo que los resultados entre ambos son muy similares y se pueden llegar a trazar relaciones entre los grupos descritos:

- *Introvert user* (Usuario introvertido): los usuarios introvertidos (18 % del total) utilizan las redes sociales como alternativa a otros medios de comunicación como el email o los servicios de chat. Es el grupo que menor uso hace de la red social, siendo inferior a una hora semanal, y que menos contenido genera. Solamente añaden a personas que conocen en el mundo real y suelen tener un círculo de amigos reducido, inferior a 50. El perfil tipo de este grupo de usuarios es un hombre de unos 45 años y suelen dejar su información de perfil a la vista de todo el mundo.
- *Novel user* (Usuario novedoso): son el segundo grupo con mayor representación, siendo 25 % de la muestra. Realizan un uso activo de la red social entrando entre una y cinco horas a la semana y generando contenido visible para sus amigos, a los que conoce del mundo real y que, igual que los usuarios introvertido, suele ser un número inferior a 50. El perfil tipo de este grupo es el de una mujer menor de 30 años y son celosos con su información, ocultando su perfil a todos los que no son sus amigos.
- *Versatile user* (Usuario versátil): el grupo de usuarios versátiles representan a casi 4 de cada 10 personas encuestadas. El uso que realizan de las redes sociales es similar al visto en el estudio anterior por los usuarios habituales. Suelen tener un grupo de amigos de tamaño medio, entre 50 y 100, y no se limitan únicamente a añadir a gente que conocen en persona. El perfil tipo de este grupo de usuarios es el de un hombre de mediana edad, entre los 36 y los 44 años, activo en la red social entrando entre una y cinco horas diarias y no necesariamente celoso con su información personal, teniendo en ocasiones el perfil público con el objetivo de conocer a nuevas personas.
- *Expert user / Communicator user* (Usuario experto / Comunicador): este grupo de usuarios es el más activo y el que más contenido genera con gran diferencia frente al resto, siendo uno de los más pequeños (19 % de los usuarios). Las personas que tienen este perfil utilizan las redes sociales como fuente principal de entretenimiento, por lo que hacen un consumo intenso de las mismas, pudiendo llegar a estar más de cinco horas diarias conectadas. Suelen tener círculos grandes de amigos, más de cien, a los que conocen principalmente en persona. El perfil tipo de este grupo es el de una mujer entre los 25 y 35 años a la que no le gusta compartir su información personal con gente que no conoce, por lo que opta por tener perfiles privados.

Conclusiones

El objetivo de este punto no era el de encontrar un modelo único con el que etiquetar a nuestros usuarios, en contraposición a lo buscado en los puntos anterior, sino tratar de encontrar unas pautas de uso entorno a criterios analizados previamente, como pueden

ser la edad o el sexo del usuario, para posteriormente tener alguna base para asignar y justificar la probabilidad de que nuestros agentes realicen o no ciertas acciones. Se ha tomado esta decisión ya que ninguno de los modelos descritos en los trabajos propuestos realizaba divisiones exactas según la edad del usuario, sino que usaban franjas de edades ambiguas y que variaban en cada propuesta. Por este motivo se ha decidido basarse en los rasgos generales que proporcionan los autores para tratar de diseñar los agentes de una manera acorde a sus propuestas, pero teniendo la libertad suficiente para estimar las probabilidades de que se realicen o no cada acción con un margen flexible.

2.5 Conclusiones

Una vez completado este apartado de la memoria se extraen las siguientes conclusiones. En primer lugar, se ha visto que ninguno de los trabajos relacionados cumple el propósito que se quiere alcanzar con este proyecto ya que la mayoría tratan de cumplir otras labores, centrándose principalmente en la automatización de tareas repetitivas, por lo que queda justificada la necesidad de desarrollo de una herramienta alternativa que cumpla el propósito planteado en este trabajo: desarrollar unos agentes que interactúen en la red social de manera orgánica, simulando el comportamiento de un usuario humano. A continuación se han analizado las principales plataformas que facilitan el trabajo y desarrollo de sistemas multiagente, llegando a la conclusión de escoger SPADE siendo el factor fundamental a la hora de tomar la decisión la experiencia previa en su integración con nuestra red social. Finalmente se han analizado los principales factores que servirán para diseñar nuestros agentes, analizando los modelos propuestos más establecidos o de vanguardia, con el objetivo de obtener un marco coherente y representativo de los usuarios tipo que podría tener una red social.

CAPÍTULO 3

Análisis del problema

En el punto anterior se han determinado algunas de las características que debe tener el producto y algunos de los sistemas que lo compondrán en base a las propuestas disponibles en la actualidad, para que el lector pueda hacerse una idea de lo que se pretende alcanzar. A continuación se definirá de manera formal toda la información relativa al sistema y software que se va a desarrollar. Esta descripción se hará en base a la estructura definida por el estándar IEEE 830 para la especificación y gestión de requisitos de un producto software, indicando únicamente los apartados que consideremos de especial relevancia.

3.1 Especificación de requisitos

3.1.1. Introducción

Propósito

En esta sección se van a tratar las diferentes especificaciones que permitirán definir y, posteriormente, implantar el sistema resultado de este trabajo: un sistema multiagente que permita simular el comportamiento de usuarios reales dentro de una red social y un plugin que permita transferir información entre la red social y el sistema multiagente.

Ámbito del sistema

El objetivo principal de la aplicación es simular la interacción de usuarios reales en la red social PESEDIA mediante el uso de agentes inteligentes. Para ello se dispondrá de un servidor web que alojará la red social y un servidor independiente en el que se instalará la plataforma de sistemas multiagente SPADE.

La implantación del sistema multiagente debe permitir, en la medida de lo posible, que pueda ser integrado en otras plataformas diferentes sin necesidad de modificar el núcleo del programa. Por tanto será necesario, a la hora de tomar decisiones durante la fase de implantación, abstraerse del comportamiento de Elgg, tratando de basarnos en APIs que permitan la comunicación entre ambos sistemas y que puedan ser transferidas de manera - relativamente - sencilla a otro ámbito.

Acorde al propósito de PESEDIA y de la legislación vigente, es vital respetar y garantizar la privacidad de los datos que se traten de manera automática por lo que, en caso de ser necesario almacenar información, se hará siempre de manera anonimizada.

El objetivo final de la herramienta será, por un lado, ayudar a los usuarios de la red social en el aprendizaje y la toma de decisiones de aspectos relacionados con la privacidad en la red y, por otro lado, proporcionar al grupo de investigación una herramienta que permita validar y probar los diferentes desarrollos y ampliaciones que se hagan en la red social.

Por lo tanto, la aplicación deberá:

- Permitir al usuario configurar el número y tipo de agentes que se deben ejecutar en base a modelos predefinidos.
- Permitir al usuario conocer el estado de los agentes mientras el sistema esté en ejecución, así como interrumpirlos cuando considere necesario.
- Simular las interacciones de usuarios reales, en base a modelos predefinidos, y publicar los resultados de dichas interacciones en PESEDIA, para que puedan ser observados por los usuarios.
- Respetar y garantizar la privacidad de la información con la que trabaja, acorde a los objetivos marcados por PESEDIA y la legislación actual.

Una vez implantada esta aplicación, los beneficios que se espera obtener son:

- Un mecanismo de generación de contenido automatizado, que permita dotar de vida a la red social sin necesidad de tener usuarios reales conectados.
- Incremento en la actividad y tiempo de uso de la plataforma por parte de los usuarios reales, debido a la sensación de incremento de tráfico e interacciones.
- Poder evaluar las herramientas de gestión de privacidad que se desarrollen en la red social en cualquier momento, sin necesidad de esperar a que hagan uso de ella los alumnos de la Escola d'Estiu.

Definiciones, acrónimos y abreviaturas

- Elgg: Plataforma de código libre que permite implementar e implantar redes sociales.
- Plug-in: Complemento para Elgg que permite añadir funcionalidad a la plataforma sin necesidad de modificar su núcleo.
- PESEDIA: Red social desarrollada por el GTI-IA en la que se implantará el producto obtenido a través de este trabajo.
- SPADE: Plataforma de código libre, basada en Python, que permite la implementación de sistemas multiagente.
- SMA: Acrónimo de Sistema Multiagente.
- API: Acrónimo del término inglés *Application Programming Interface*, Interfaz de Programación de Aplicaciones en castellano.

Visión general del documento

Para el correcto desarrollo de este proyecto es necesario definir las líneas generales del sistema junto a una serie de requisitos, que son los temas a tratar en las siguientes secciones. En la sección 3.1.2 se definirá la perspectiva y funcionalidad del producto, incluyendo los principales actores que intervengan en el mismo y las posibles restricciones y dependencias a tener en consideración. En la sección 3.1.3 se hará una descripción detallada de los diferentes requisitos funcionales y de rendimiento propios del sistema, así como las posibles restricciones en el diseño y sus diferentes atributos (fiabilidad, mantenibilidad, portabilidad, etc.).

3.1.2. Descripción general

Perspectiva del producto

El sistema a desarrollar se divide en dos subsistemas independientes, pero a la vez complementarios:

- El primer subsistema pondrá a disposición de los administradores de la red social una ventana de configuración, accesible vía web, a través del menú de administración de PESEDIA. Este primer subsistema se diseñará gracias al soporte que ofrece Elgg para el desarrollo de extensiones.

Gracias a esta ventana de configuración, el administrador podrá añadir y retirar agentes del sistema multiagente, podrá decidir el comportamiento esperado de los agentes - utilizando una serie de parámetros de configuración predefinidos - y podrá consultar el estado de los agentes para observar si existe alguna anomalía.

Además, este subsistema incorporará los mecanismos necesarios para que los agentes puedan obtener información de la red social de manera autónoma. Para lograrlo se implementará una API que se encargue de extraer la información de PESEDIA y que pueda ser accesible por otros sistemas.

Este primer subsistema será una ampliación de la red social y hará uso de PHP como lenguaje de desarrollo, Apache como servidor web y las funciones que ofrece Elgg como mecanismo para acceder a la información de PESEDIA.

- El segundo subsistema consistirá en un sistema multiagente diseñado con la plataforma SPADE en el que se realizarán las simulaciones necesarias, acorde a la configuración realizada por el usuario en el primer subsistema.

Este segundo subsistema, a pesar de diseñarse para funcionar inicialmente en PESEDIA, debe tener las características necesarias para poder ser integrado en otras redes sociales, como podrían ser Facebook o Twitter, con la mayor transparencia posible, por lo que debe evitarse incluir peculiaridades que se puedan encontrar únicamente en PESEDIA y que imposibiliten su uso con otras plataformas.

Este subsistema se desarrollará utilizando Python, ya que es el lenguaje empleado por SPADE.

Funciones del producto

En la figura 3.1 se representan de manera gráfica las distintas funcionalidades que tendrá el sistema junto a los agentes que pueden/deben intervenir en las mismas. Para ampliar la información mostrada en la figura 3.1 puede consultarse la información incluida en la tabla 3.1 en la que se describen las características y funcionalidades que deben cubrirse en cada caso de uso, proporcionando una vista general de lo que se espera cumplir una vez implementado e implantado el sistema.

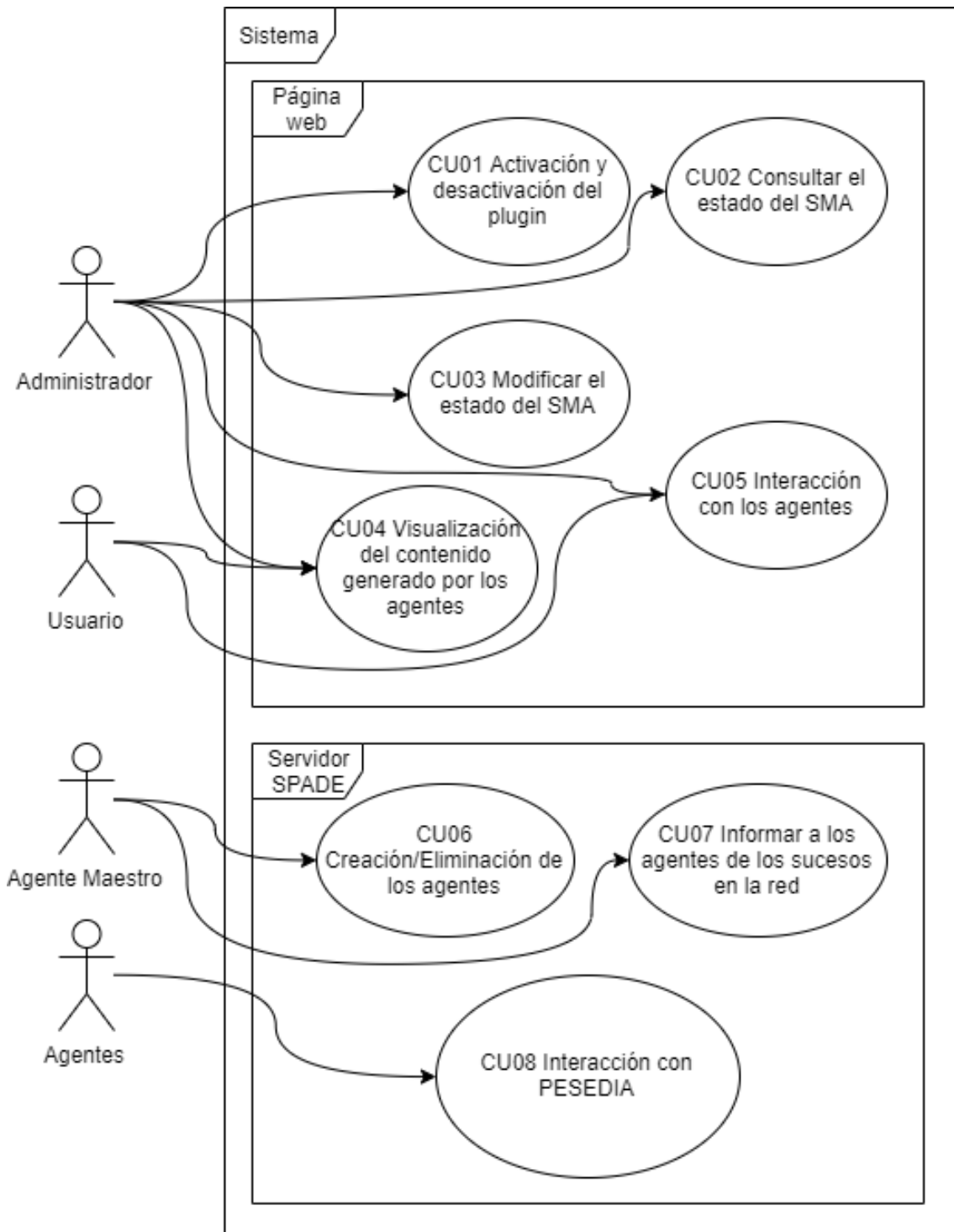


Figura 3.1: Diagrama de casos de uso

Caso de uso	Descripción
CU01 Activación y desactivación del plugin	El administrador debe tener la capacidad de activar y desactivar el plugin que permite la interacción entre PESEDIA y el SMA cuando considere.
CU02 Consultar el estado del SMA	Una vez activado el plugin, el administrador debe poder visualizar el estado del SMA desde el panel de administración de la web para comprobar su correcto funcionamiento.
CU03 Modificar el estado del SMA	El administrador debe tener la capacidad de añadir y retirar agentes del sistema a conveniencia. Además debe ser capaz de definir el comportamiento de los agentes a partir de unos parámetros base.
CU04 Visualización del contenido generado por los agentes	Tanto los usuarios como los administradores deben poder visualizar el contenido generado por los agentes, de la misma manera que verían el contenido generado por un humano.
CU05 Interacción con los agentes	Tanto los usuarios como los administradores deben poder interactuar con los agentes, ya sea contestando a los mensajes que estos publiquen, pudiendo añadirles como amigos, etc.
CU06 Creación/Eliminación de los agentes	El agente maestro debe tener la capacidad de crear agentes secundarios en base a la configuración que el administrador realice en la página web.
CU07 Informar a los agentes de los sucesos en la red	El agente maestro debe tener la capacidad de comunicar a los agentes secundarios todos los sucesos relevantes que acontezcan en PESEDIA.
CU08 Interacción con PESEDIA	Los agentes secundarios deben ser capaces de extraer, por su cuenta, información de PESEDIA, así como de generar y guardar información en ella.

Tabla 3.1: Descripción de los casos de uso

Características de los usuarios

En los casos de uso se han identificado cuatro tipos de usuarios diferentes que pueden participar en el sistema, dos de ellos reales (administradores y usuarios) y otros dos ficticios (los agentes). En este apartado se detalla, tal y como se puede observar en la tabla 3.2, las diferencias generales entre los dos tipos de usuario y las actividades en las que participan.

Tipo de usuario	Descripción	Actividades
Administrador	Persona real responsable del mantenimiento y configuración de la red social. Es el encargado de la configuración, puesta en marcha y detención del plugin que permite la comunicación con el sistema multiagente.	CU01 - Activación y desactivación del plugin CU02 - Consultar el estado del SMA CU03 - Modificar el estado del SMA CU04 - Visualización del contenido generado por los agentes CU05 - Interacción con los agentes
Usuario	Cualquier persona real, miembro de la red social, que no figura como Administrador. Puede entrar a la red social e interactuar con el sistema multiagente a través de los mecanismos proporcionados por Elgg.	CU04 - Visualización del contenido generado por los agentes CU05 - Interacción con los agentes
Agente maestro	Usuario ficticio que representa al agente coordinador del SMA. Su labor principal es realizar de interfaz entre la red social y los agentes del SMA, exponiendo al exterior las funciones propias de SPADE.	CU06 - Creación/Eliminación de los agentes CU07 - Informar a los agentes de los sucesos en la red
Agentes	Usuarios ficticios que, a través de su interacción con PESEDIA, dotarán a la red social de actividad.	CU08 - Interacción con PESEDIA

Tabla 3.2: Tipos de usuarios y sus características

Restricciones

El sistema sufre de una serie de restricciones causadas por su naturaleza como producto software distribuido vía web. En la tabla 3.3 se detallan estas restricciones.

Restricción	Descripción
RS01 Docker	El sistema utilizará Docker para gestionar la infraestructura, por lo que será necesario disponer de un host compatible para hacer el despliegue del mismo. El sistema es funcional sin utilizar Docker, pero será responsabilidad de la persona encargada del despliegue garantizar la correcta configuración de los diferentes subsistemas.
RS02 Conexión a Internet	Para que el administrador pueda hacer uso del plugin ofrecido en el primer subsistema es necesario que el host en el que se encuentre el sistema disponga de conexión a Internet. De igual manera los usuarios necesitarán acceder a la red social para interactuar con los agentes.
RS03 Privacidad	PESEDIA es una red social enfocada a enseñar a sus usuarios la importancia de tener unos patrones de uso responsables. Para alinear el producto con el espíritu del proyecto, es necesario que todos los posibles medios de comunicación que surjan a partir de esta ampliación sean, en todo momento, respetuosos con el contenido de los mensajes que se vayan a tratar, garantizando en la medida de lo posible el anonimato de sus autores.
RS04 Edad de los usuarios	Al tratarse de una red social pensada para ser usada por adolescentes, es necesario garantizar que el contenido generado por los agentes tome en consideración esta problemática, impidiéndoles publicar información no apropiada.
RS05 Protocolos de comunicación	Como se van a integrar dos plataformas diferentes, Elgg y SPADE, es necesario respetar los protocolos de comunicación propuestos por cada plataforma, aumentando la complejidad de la implementación.
RS06 Lenguajes de programación	Debido a las limitaciones generadas por las plataformas empleadas es necesario utilizar los lenguajes de programación usados en cada una de ellas. PHP en el caso del primer subsistema y Python en el caso del segundo.

Tabla 3.3: Restricciones del sistema

Suposiciones y dependencias

Para el correcto funcionamiento del sistema se asume que se cumplen las siguientes dependencias:

Subsistema 1 - Plugin Web

- Existe una red social funcional que utilice la plataforma Elgg en su versión 2.3. No se garantiza el funcionamiento del plugin en versiones más recientes debido a algunos cambios en las funciones que permiten el acceso a los datos.
- Se ha habilitado en la red social el plugin "Web Services", que permite las llamadas a la API.
- Se ha habilitado en la red social el plugin "API Admin" (o algún otro similar) que permita generar y gestionar claves para poder conectar el Sistema Multiagente a PESEDIA a través de la API.
- Se asume que la máquina es capaz de enviar y recibir llamadas utilizando los protocolos XMPP y HTTP/HTTPS.

Subsistema 2 - Sistema Multiagente

- Se dispone de una máquina capaz de ejecutar programas en Python 3.7. No se puede garantizar el funcionamiento con versiones más recientes de Python debido a las librerías que se utilizan.
- Se asume que la máquina es capaz de enviar y recibir llamadas utilizando los protocolos XMPP y HTTPS.

- Se asume que durante la configuración del SMA se proporcionará una clave para poder conectarse a la API de PESEDIA.

Las asunciones derivadas del uso de versiones específicas de software se solucionan utilizando la infraestructura Docker propuesta. En caso de querer optar por hacer uso de esta plataforma para el despliegue del sistema surge una nueva necesidad:

- Disponer de una máquina capaz de levantar contenedores Docker. En los puntos siguientes se encuentran los enlaces a los requisitos que tiene Docker en los sistemas operativos más populares:
 - Requisitos en Windows ¹
 - Requisitos en Mac ²
 - Requisitos en Linux ³

3.1.3. Requisitos específicos

En este apartado se van señalar todos los requisitos que debe cumplir el sistema con el objetivo de poder considerarlo como válido. Al contener dos subsistemas el protocolo seguido para identificar cada requisito ayuda a identificar a que subsistema pertenece, siendo el prefijo S1-* el empleado para identificar los requisitos relacionados con el plugin web y S2-* el empleado para los requisitos correspondientes al sistema multiagente. En caso de que algún requisito sea imprescindible en ambos subsistemas no se añadirá el prefijo en el identificador. Con el objetivo de facilitar la lectura del documento se agruparán por apartados los requisitos según al subsistema al que pertenezcan.

Interfaces externas

Identificación	S1-RI01
Título	Adaptación al dispositivo
Descripción	El menú que se ofrece al usuario vía web debe adaptarse al tamaño de pantalla del dispositivo cliente y, de igual manera, debe ofrecer las mismas oportunidades a la hora de interactuar con él.
Prioridad	Media

Tabla 3.4: Requisito S1-RI01

Identificación	S1-RI02
Título	Interfaz de configuración simple
Descripción	La configuración del sistema multiagente puede llegar a ser extremadamente compleja por lo que se debe tratar de abstraer al usuario en la medida de lo posible, reduciendo la cantidad de información a introducir, delegando en el propio plugin la toma de decisiones menos relevantes.
Prioridad	Media

Tabla 3.5: Requisito S1-RI02

¹<https://docs.docker.com/docker-for-windows/install/#what-to-know-before-you-install>

²<https://docs.docker.com/docker-for-mac/install/#what-to-know-before-you-install>

³<https://docs.docker.com/install/>

Funciones

Identificación	S1-RF01
Título	Activación/Desactivación del plugin
Descripción	La activación y desactivación del plugin implica una conexión y desconexión con el SMA, por lo que debe garantizarse la estabilidad del sistema.
Prioridad	Alta

Tabla 3.6: Requisito S1-RF01

Identificación	S1-RF02
Título	Añadir/Retirar agentes
Descripción	El administrador debe poder añadir y retirar agentes del sistema sin que esto provoque errores en la red social.
Prioridad	Alta

Tabla 3.7: Requisito S1-RF02

Identificación	S1-RF03
Título	Configurar agentes según modelos
Descripción	El administrador debe poder configurar en cierta medida el comportamiento de los agentes en base a unas opciones predefinidas.
Prioridad	Alta

Tabla 3.8: Requisito S1-RF03

Identificación	S1-RF04
Título	Visualizar el estado de la red
Descripción	El administrador debe poder visualizar el estado del SMA desde el propio plugin.
Prioridad	Media

Tabla 3.9: Requisito S1-RF04

Identificación	S1-RF05
Título	Interactuar con los agentes
Descripción	Cualquier usuario debe poder visualizar contenido generado por los agentes y debe poder interactuar con ellos en base a las acciones soportadas.
Prioridad	Alta

Tabla 3.10: Requisito S1-RF05

Identificación	S1-RF06
Título	Envío de información al SMA
Descripción	El plugin debe enviar al SMA toda la información de interés que ocurra en la red social.
Prioridad	Alta

Tabla 3.11: Requisito S1-RF06

Identificación	S2-RF01
Título	Extracción/Inserción información en PESEDIA
Descripción	Los agentes deben ser capaces de extraer de manera autónoma información de la red social y de guardar el contenido que quieran generar a partir de sus acciones.
Prioridad	Alta

Tabla 3.12: Requisito S1-RF07

Requisitos de rendimiento

Identificación	RR01
Título	Disponibilidad
Descripción	Para considerar que el servicio ofrece alta disponibilidad, ambos subsistemas deben ser accesibles el 99.99 % del tiempo y se debe garantizar que en caso de fallo se informe correctamente a la persona responsable.
Prioridad	Alta

Tabla 3.13: Requisito RR01

Identificación	RR02
Título	Tiempo de respuesta
Descripción	El tiempo máximo de computación de los diferentes componentes del sistema debe ser de 1 segundo por operación, este límite solamente puede rebasarse en caso de fallos en la infraestructura, no producidos por el propio programa.
Prioridad	Alta

Tabla 3.14: Requisito RR02

Identificación	RR03
Título	Sobrecarga de la red
Descripción	El funcionamiento de los agentes no debe producir un exceso de carga en el servidor que aloje la red social. En caso de penalizar el normal funcionamiento de la misma - el uso de recursos disponibles sobrepase el 70 % - debe informarse al responsable para que tome las acciones necesarias. En caso de superar el 90 % se detendrá la comunicación con el sistema multiagente para evitar la caída de la aplicación.
Prioridad	Alta

Tabla 3.15: Requisito RR03

Identificación	S1-RR01
Título	Número de conexiones
Descripción	Durante los talleres de privacidad en redes sociales, en los que los alumnos de L'Escola d' Estiu hacen uso de PESEDIA, es cuando suceden los momentos de máxima carga en la red, pudiendo alcanzar una media de 125 usuarios conectados de manera simultánea. Por tanto debe garantizarse que los agentes no producirán sobrecarga en la red, permitiendo que, a las 125 conexiones generadas por los usuarios, se sumen otras 125 que correspondan a los agentes, alcanzando un nivel de garantía de 250 conexiones simultáneas en total.
Prioridad	Alta

Tabla 3.16: Requisito RR04

Restricciones de diseño

Identificación	RD01		
Título	Protocolos de comunicación		
Descripción	Debido a las especificaciones de las plataformas utilizadas es necesario adaptar el sistema para que permita comunicación empleando diferentes protocolos.		
Prioridad	Alta	Media	Baja

Tabla 3.17: Requisito RD01

Atributos del sistema

- **Fiabilidad:** el sistema debe cumplir los requisitos de rendimiento expuestos previamente en este documento.
- **Mantenibilidad:** para garantizar que en el futuro se pueda continuar el desarrollo de este sistema, así como facilitar la resolución de posibles contingencias, deben tomarse las siguientes medidas.
 - **Prevención:** deben seguirse las guías de estilo propias de cada lenguaje empleado en el desarrollo de los diferentes subsistemas. Así mismo deben utilizarse protocolos estandarizados y propuestos por las plataformas/frameworks utilizados.
 - **Corrección:** debe informarse al responsable del sistema ante cualquier comportamiento no esperado, para que este pueda tomar las consideraciones necesarias. El sistema debe ser consistente por lo que, en caso de ser necesaria su detención, debe garantizarse la seguridad de la información.
- **Portabilidad:** el sistema debe ser capaz de funcionar en cualquier máquina que cumpla los requisitos indicados en esta especificación. Se propone el uso del sistema de contenedores Docker.
- **Seguridad:** el sistema debe garantizar que la información que se transmita a través del mismo sea seguro, utilizando los protocolos de comunicación necesarios. Se debe garantizar que solamente los usuarios autorizados puedan realizar modificaciones en el mismo, utilizando para ello los protocolos de autenticación ofrecidos por Elgg. Se debe asegurar que los diferentes componentes no son accesibles de manera no prevista, utilizando claves SSH.

3.2 Análisis del marco legal y ético

La herramienta resultado de este trabajo necesita adquirir información de una red social y, por tanto, tendrá acceso a información vertida por usuarios reales. En concreto la información se va a obtener de la red social PESEDIA, empleada principalmente por los alumnos de la Escola d'Estiu organizada por la UPV. El sistema multiagente hará un tratamiento de la información automático y, por tanto, hay algunas consideraciones que deben realizarse.

Por un lado, los agentes tendrán acceso a toda la información pública (visible para cualquier usuario registrado) contenida en la red social. Por ello podrán visualizar, entre otros datos, los identificadores de los usuarios generados en la red social y sus nombres públicos, la información de los perfiles de usuario con visibilidad pública, los mensajes de texto/imagen/vídeo de los diferentes muros que tengan visibilidad pública, los

grupos de los que puedan formar parte y el contenido de los mismos, etc. En resumen, podrán obtener toda la información que los usuarios quieran compartir, bien con cualquier usuario de manera pública, o bien con los agentes a través de comunicación directa entre ambos.

Contrariamente, los agentes no tendrán acceso a ningún tipo de información compartida de manera privada entre los usuarios o a través de cualquier sistema dentro de la red social que quede fuera del alcance de cualquier otro usuario. Adicionalmente, los agentes no almacenarán información (al menos no en esta primera versión) del contenido que se genere en la red social y por tanto, para garantizar la privacidad de la información que se transfiera entre PESEDIA y los agentes, será únicamente necesario encriptar los mensajes cuando vayan a enviarse entre las plataformas.

En la actualidad los padres o tutores legales de los usuarios de la red social deben firmar una hoja de consentimiento para que los alumnos - menores de edad - puedan hacer uso de la plataforma. En esa autorización se considera, entre otros, la autorización para el tratamiento automatizado de los datos publicados en la red social. Para garantizar que la información sea anónima, se proporciona a los alumnos que van a participar en la red una serie de usuarios ficticios y las únicas personas con acceso a la relación entre alumno y usuario son sus monitores de grupo. No obstante, una vez implantado el sistema multi-agente, sería recomendable incluir en la autorización una opción que permita a los padres y tutores de los alumnos excluir a sus tutelados de esta nueva opción, informando de los pros y contras que tendrá en la formación del alumno en caso de optar a ella.

Teniendo en cuenta estas limitaciones y sugerencias puede considerarse que la herramienta resultado de este trabajo se alinea con el espíritu de PESEDIA y cumple la legislación vigente en cuanto a normativa de protección de datos, Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales [18], comúnmente denominada como RGPD (*Reglamento general de protección de datos*).

3.3 Análisis de riesgos

En este apartado vamos a analizar los diferentes riesgos inherentes al desarrollo y puesta en marcha de esta aplicación. En primer lugar se analizarán los riesgos que puedan afectar al usuario y en segundo lugar los riesgos que puedan afectar a la integración con el sistema actual. Los riesgos de usuario se identificarán con el prefijo RUsu* y los riesgos de integración sistema con el prefijo RInt*.

Identificación	RUsu01
Título	El comportamiento de los agentes es poco orgánico.
Descripción	El comportamiento de los agentes no es natural, el contenido que publican no tiene sentido.
Impacto	Los usuarios pueden percibir que los agentes son organismos artificiales y por ello pueden decidir ignorar sus mensajes o incluso abandonar la página.
Medidas	Generación de modelos que permitan representar el comportamiento de un humano. Limitación de interacciones con la red social.

Tabla 3.18: Riesgo RUsu01

Identificación	RUsu02
Título	Exceso de actividad en la red social
Descripción	La red social tiene más actividad de la esperada para su número de usuarios.
Impacto	El usuario percibe una actividad anormal, pudiendo llegar a sentirse saturado por el exceso de contenido, reduciendo su participación.
Medidas	Uso de algoritmos matemáticos para "racionalizar" la participación de los agentes.

Tabla 3.19: Riesgo RUsu02

Identificación	RInt01
Título	Saturación y caída del sistema
Descripción	Un exceso de conexiones generadas por agentes puede provocar la caída del sistema.
Impacto	No disponibilidad del servicio.
Medidas	Mecanismos de aviso a los administradores. Gestión "elástica" (eliminación y/o creación según recursos) de los agentes.

Tabla 3.20: Riesgo RInt01

Identificación	RInt02
Título	Gestión de infraestructura externalizada
Descripción	La gestión de la infraestructura es realizada por técnicos ajenos al proyecto. No asignación del número de recursos necesario o asignación incorrecta de recursos.
Impacto	No disponibilidad del servicio, funcionamiento incorrecto/poco fluido.
Medidas	Mejoras en los mecanismos de comunicación y coordinación con el equipo técnico

Tabla 3.21: Riesgo RInt02

Identificación	RInt03
Título	Conflictos con otros plugin
Descripción	PESEDIA está en constante desarrollo, por lo que en el futuro es posible necesitar un plugin que no funcione junto al servicio actual.
Impacto	No funcionamiento de los nuevos desarrollos. No disponibilidad del servicio.
Medidas	Encapsulación del plugin para que tenga su propio mecanismo de fallos y alertas.

Tabla 3.22: Riesgo RInt03

Identificación	RInt04
Título	Conflicto con otros SMA ofrecidos en la plataforma
Descripción	PESEDIA dispone de otros SMA que podrían llegar a entrar en conflicto con el desarrollado en este proyecto, produciendo fallos y/o sobrecargando la red.
Impacto	No disponibilidad del sistema.
Medidas	Estudiar el desarrollo del resto de SMA e intentar integrarlos en un único sistema.

Tabla 3.23: Riesgo RInt04

3.4 Identificación y análisis de soluciones posibles

En el capítulo anterior (Estado del arte) se propusieron, analizaron y seleccionaron las diferentes propuestas más relevantes, tanto para la plataforma sobre la que realizar el desarrollo del sistema multiagente, como para el modelado de los propios agentes. En el siguiente capítulo se detallarán todas las decisiones y consideraciones correspondientes al desarrollo del sistema por lo que, nuevamente, se emplaza al lector a su consulta.

A modo de resumen se recuerda que se ha seleccionado la plataforma SPADE para el desarrollo del sistema multiagente y el uso de la API de PESEDIA. Esto implica que deban utilizarse dos vías de comunicación: XMPP y HTTP/HTTPS. Para el modelado de los agentes se ha generado una propuesta nueva a través de la combinación de los diferentes modelos propuestos en los trabajos más relevantes en la actualidad.

3.5 Conclusiones

En este capítulo se ha realizado el análisis y especificación de requisitos que debe cumplir el sistema para su aceptación. Se han observado las diferentes implicaciones legales que pueden producirse a raíz de su desarrollo, los diferentes riesgos inherentes al mismo y finalmente se han condensado las consideraciones correspondientes a las soluciones propuestas, analizadas en otros capítulos de esta memoria.

Gracias a este análisis, y a las consideraciones extraídas durante el estudio del estado del arte, se han podido detectar todos los puntos críticos que ayudarán a la consecución de este proyecto, incluyendo las principales innovaciones, problemáticas y objetivos principales y secundarios.

CAPÍTULO 4

Solución propuesta

Una vez identificados los requisitos indicados en el anterior capítulo, es necesario abordar el diseño del sistema que permitirá la consecución de los mismos.

En primer lugar se presentará al lector una visión global del sistema, en la que se analizarán de manera informal los puntos principales que ayuden a comprender el funcionamiento de la solución a implementar. En este mismo punto se irán detallando adicionalmente las diferentes tecnologías que se utilizarán en cada parte del sistema.

En el segundo punto se detallarán, de manera formal utilizando UML, las diferentes capas del sistema. En el tercer punto se expondrán los diferentes modelos para los agentes que se encontrarán en la red social, propuestos como resultado del estudio realizado en el capítulo 2.

En los últimos punto se detallarán, en base a la información extraída tanto en el capítulo anterior como en el actual, los diferentes recursos: materiales, temporales y humanos, que se necesitarán para la consecución de este proyecto.

4.1 Especificación conceptual

4.1.1. Visión global

Para comprender la estructura del proyecto es necesario tener una visión general del mismo: componentes que forman el sistema, mecanismos de comunicación entre ellos, tecnologías necesarias, etc. En la figura 4.1 se muestran dichos componentes del sistema junto al flujo de comunicaciones de los mismos. A continuación de la imagen se explica el funcionamiento de los diferentes canales de comunicación utilizando el número indicado en la imagen como referencia.

El flujo de trabajo del sistema se divide en cuatro partes diferenciadas, la primera de ellas surge de la interacción entre el administrador y el plugin web (el subsistema 1 acorde a la especificación de requisitos). La segunda y tercera parte incluyen los protocolos y vías de comunicación empleados por el sistema multiagente (el subsistema 2 acorde a la especificación de requisitos) y el plugin web. El cuarto y último paso corresponde a la interacción entre usuarios y página web.

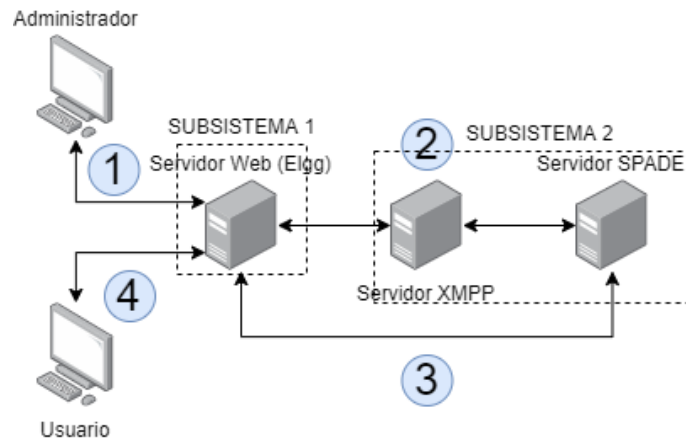


Figura 4.1: Diagrama conceptual del sistema

1. El primer punto crítico trata de resolver el problema que surge ante la necesidad de comunicar al administrador de PESEDIA con el sistema multiagente. Para solventar este escollo se ofrecerá al administrador un plugin web que incluirá tanto las interfaces como los mecanismos de comunicación necesarios para hacer efectivo el enlace entre PESEDIA y el sistema multiagente.

Tratando de explicar de manera conceptual los pasos a realizar en este punto debemos asumir que el administrador se conectará vía web a través del navegador incluido en su dispositivo (PC, móvil, tablet, etc.) al panel de administración de PESEDIA y activará el plugin proporcionado desde el menú correspondiente. Para el desarrollo del proyecto este proceso de activación del plugin se puede entender como una caja negra, ya que la activación, desactivación y gestión en general de plugins es una funcionalidad que ya se encuentra en Elgg antes de realizar esta ampliación y por tanto no es necesario comprender/explicar exhaustivamente el proceso.

Una vez activado el plugin el administrador encontrará una nueva opción dentro del menú de administración (detallado gráficamente en el apartado 4.2.1) y, al acceder a esta nueva opción, tendrá la posibilidad de realizar una conexión al sistema multiagente. Cuando se trate de realizar la conexión, el plugin web consultará al sistema multiagente su estado para saber si está disponible y, en caso de estarlo, mostrará al usuario una nueva vista en la que podrá observar el estado del sistema, así como las diferentes operaciones posibles: añadir y retirar agentes, desconectarse del sistema, etc.

Una vez que el administrador termine su trabajo podrá desconectar la sesión. Esta desconexión indica al plugin que no necesita solicitar más información al sistema multiagente, que seguirá en funcionamiento y los agentes creados seguirán interactuando con la red con total normalidad. En caso de querer eliminar a los agentes para que no sigan generando contenido, el administrador puede optar por retirarlos todos desde el menú de configuración (como se muestra en los fotomontajes del apartado 4.2.1), o bien podrá desactivar el plugin al completo.

2. El segundo punto trata uno de los mecanismos de comunicación empleado entre PESEDIA y el sistema multiagente, que será explicado con mayor detenimiento en el siguiente apartado. La plataforma que vamos a utilizar para el desarrollo del sistema multiagente es SPADE. Esta plataforma utiliza el protocolo XMPP como mecanismo de comunicación de los agentes, por tanto es lógico tratar de utilizar este protocolo para gestionar las comunicaciones en nuestro sistema.

Brevemente, XMPP (*Extensible Messaging and Presence Protocol*) es un protocolo basado en XML e ideado para facilitar la mensajería instantánea. Para ello el sistema dispondrá de varios identificadores JID (*Jabber ID*) que simularán ser usuarios de una plataforma de mensajería. Estos usuarios serán asignados a los diferentes actores del sistema y se encargarán de enviar, recibir e interpretar los diferentes mensajes enviados por la red a modo de mensaje personal.

Adicionalmente, y aunque no es el objetivo de este apartado conocer en detalle la estructura del sistema (profundizaremos en la sección 4.1.2), antes de continuar con esta visión global necesitamos saber que en el sistema multiagente tendremos un agente coordinador, denominado agente maestro, que se encargará de gestionar a todos los diferentes agentes que introduzcamos en el sistema. Por tanto tanto el plugin web, como cada uno de los agentes del sistema multiagente dispondrán de su propio JID. Además los agentes dispondrán también de un usuario en la red social, ya que sin él no podrían interactuar con PESEDIA.

Una vez realizados estos dos apuntes procedemos con la explicación del segundo punto.

Cuando el administrador realice alguna operación susceptible de necesitar recibir o enviar información al sistema multiagente - consultar su estado, por ejemplo - siguiendo las pautas indicadas en el punto 1, el plugin se encargará de enviar la solicitud, empleando su JID, al JID asociado al agente maestro. Este agente maestro procesará el mensaje recibido, comunicará a los agentes la información relevante para ellos y responderá al plugin web en consecuencia. El plugin web procesará la respuesta y actualizará la información necesaria en pantalla para que el administrador pueda conocer el resultado de la operación.

3. El tercer punto trata otro de los mecanismos de comunicación empleados entre PESEDIA y el sistema multiagente. Cada vez que el agente maestro notifique a los agentes de un suceso de su interés o cuando ellos mismos lo consideren, deben disponer de una vía de comunicación que les permita realizar las operaciones necesarias para satisfacer su objetivo.

Elgg ofrece al programador la posibilidad de implementar funciones accesibles a través de una API, como forma nativa para gestionar la comunicación con sistemas externos. Por tanto, parece lógico pensar que los agentes utilicen este mecanismo de comunicación para interactuar con la red social y poder tanto extraer como persistir información de la misma sin necesidad de pasar por el agente maestro.

Los agentes pueden tener un objetivo que requiera extraer información de PESEDIA. Para lograrlo consultarán vía API el *endpoint* (una dirección web específica) diseñado para esta tarea. Cuando obtengan la respuesta de la API podrán realizar las operaciones necesarias para intentar cumplir su objetivo y, en caso de ser necesario, tratarán de persistir la información de nuevo en la red social utilizando otro *endpoint* diferente. Para poder interactuar con la API los agentes deberán indicar el usuario que tienen asignado en la red social.

4. Este último punto trata de resolver la manera en la que PESEDIA (en concreto el plugin web) enviará información al sistema multiagente para que sea consciente de los cambios ocurridos en el sitio web a partir de la interacción de los usuarios y otros agentes. Cada vez que los agentes persistan información en PESEDIA es probable que dicha información deba publicarse para que tengan conocimiento de la misma los usuarios reales de la red social u otros agentes del sistema.

Tanto cuando sean los agentes los que publiquen contenido vía la API, como los usuarios lo hagan desde las opciones pertinentes disponibles dentro de la red so-

cial, necesitamos que tanto unos como otros dispongan de la información necesaria. Para solventar la comunicación agente → usuario (entiéndase usuario como un usuario real u otro agente) ya hemos explicado que dispondremos de varias funciones expuestas dentro de la API que permitirán persistir la información. Esta información una vez se encuentre en la base de datos será tratada por Elgg automáticamente y el propio framework facilitará la visibilidad de la misma en base a lo indicado.

Para poder solventar la comunicación usuario → agente (entiéndase nuevamente usuario como un usuario real u otro agente) haremos uso de los mecanismos de Elgg que nos permiten capturar los diferentes eventos asociados a las acciones que suceden en la red social. Cuando se dispare uno de estos eventos procederemos a enviar vía XMPP (como se explicó en el paso 2) la información al agente maestro y éste se encargará de difundirla a los agentes según considere apropiado.

4.1.2. Comunicaciones entre los subsistemas

En este apartado vamos a tratar de desgranar algunos de los conceptos explicados de manera conceptual en el apartado anterior en lo referente a los mecanismos de comunicación propuestos para el sistema a desarrollar. Por ahora vamos a abstraernos de los componentes hardware que compondrán el sistema y vamos a centrarnos en los diferentes componentes software del mismo. En la figura 4.2 podemos ver los dos principales subsistemas.

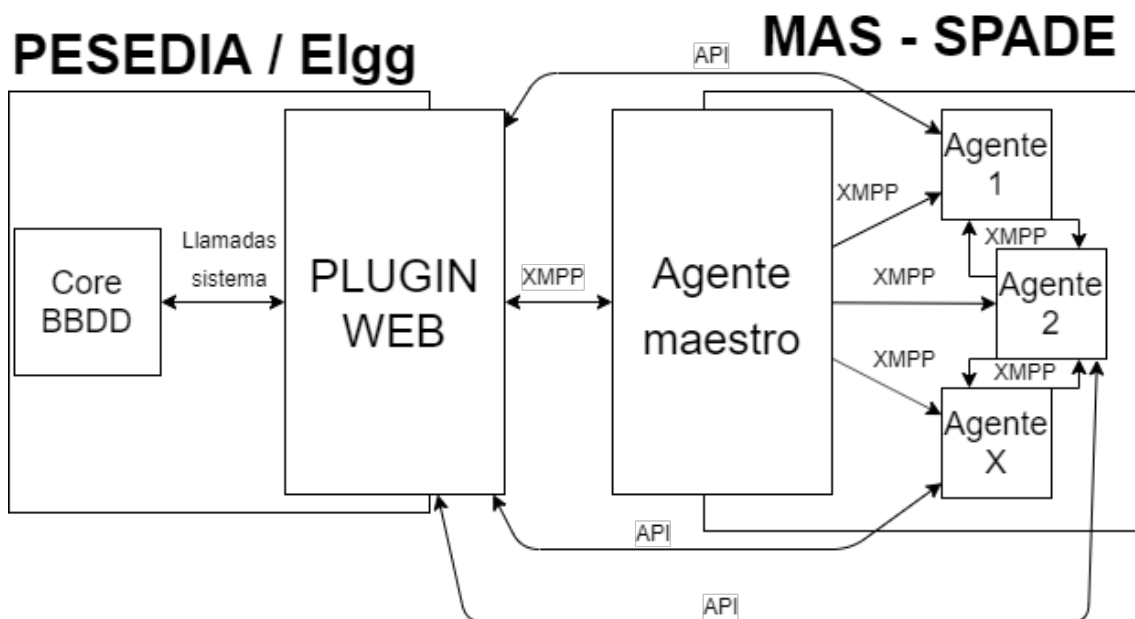


Figura 4.2: Mecanismos de comunicación empleados por el sistema

A la izquierda de la figura 4.2 puede verse la red social PESEDIA, en la que destacan el plugin web, obtenido como resultado de este proyecto, que será el responsable de enlazar las comunicaciones entre el núcleo (Core) de Elgg y el sistema multiagente. Este plugin web ofrecerá las herramientas necesarias para abstraer al administrador de las complejidades que existen a la hora de enlazar los dos sistemas, proporcionando la funcionalidad necesaria para que pueda ser expandido en función de las necesidades actuales y futuras.

Para cumplir su objetivo el plugin web hace uso de XMPP para gestionar la comunicación con el agente maestro y expande la API de Elgg para que se puedan comunicar

el resto de agentes. La comunicación en ambos casos es bidireccional, pero el origen de las peticiones es único. Esto quiere decir que la vía de comunicación entre el plugin y el agente maestro permite tanto enviar como recibir comunicaciones, aunque siempre será el plugin web el responsable de indicar al agente maestro lo que espera, y no al contrario. En el caso de las llamadas vía API sucede lo mismo, se puede tanto enviar como recibir información pero en este caso el origen de las peticiones se encuentra en los diferentes agentes, excluyendo al maestro.

Por esta vía de comunicación enviaremos todos los mensajes que contengan información relevante para el sistema multiagente. A continuación se detalla uno de los flujos de trabajo que ejemplifican estas comunicaciones por XMPP.

Añadir o retirar agentes del sistema:

1. El administrador indicará al plugin web que desea añadir agentes.
2. El plugin procederá a crear los usuarios necesarios en la red social vía llamadas con funciones del sistema.
3. El plugin comunica al agente maestro que debe crear agentes y le indica los usuarios que debe asignar.
4. El agente maestro procede a la creación de los agentes y les indica cuál es su usuario en la red social.
5. El agente maestro comunica al plugin el resultado.
6. El plugin comunica al administrador el resultado.

En el sistema multiagente, situado a la derecha en la figura 4.2, destacan varios elementos. En primer lugar el agente maestro, responsable de coordinar y dirigir (sin limitar la autonomía) a los agentes. Este agente maestro, como ya hemos visto, es el enlace entre el sistema multiagente y PESEDIA. Conecta con PESEDIA a través de una vía bidireccional, aunque como ya se ha comentado solamente podrá enviar información cuando sea solicitada por el plugin, nunca por decisión propia. Para comunicarse con los agentes utilizará una vía de comunicación unidireccional, por la que volcará la información necesaria para que los agentes tengan conocimiento de ella.

Por otro lado los agentes que conforman el sistema multiagente, excluyendo al maestro, tendrá comunicación con la red social vía API para poder extraer o introducir información en la misma bajo su propio criterio, sin necesidad de limitar su actuación a estímulos externos, consiguiendo así un mecanismo de comunicación que les permita ser proactivos. Vamos a ver un caso en el que los agentes pueden, de manera autónoma, extraer información de la red.

Añadir un amigo:

1. Uno de los agentes determina que quiere añadir un amigo.
2. El agente solicita vía API la información de usuarios de la red.
3. Una vez que el agente dispone de la información correspondiente la analiza y decide a cuál de los usuarios quiere mandar una solicitud.
4. Nuevamente vía API el agente manda la solicitud de amistad.

Si queremos que los agentes puedan ser reactivos será necesario que en algunas operaciones se exploten las dos vías de comunicación. Finalmente, vamos a ilustrar un caso

más complejo en el que se explotarán los dos protocolos de comunicación, XMPP y API, para conseguir sacar el máximo potencial al sistema.

Creación de mensajes y respuesta de los agentes. Parte 1 - XMPP:

1. El usuario (entiéndase por usuario un usuario real u otro agente) publica contenido en la red social.
2. El plugin comunica al agente maestro la creación del contenido, adjuntando la información necesaria.
3. El agente maestro comunica a todos los agentes de la red la creación del contenido para que tengan constancia de ello.
4. El agente maestro informa al plugin que ha comunicado a todos los agentes la información.

Creación de mensajes y respuesta de los agentes. Parte 2 - API:

1. Los agentes han recibido la información de lo sucedido en la red social vía XMPP como se ha indicado en la parte 1.
2. Los agentes analizan el contenido del mensaje.
3. Una vez analizado el contenido del mensaje, los agentes deciden si quieren responder al mismo. Si quieren responder continuarán el proceso, en caso contrario pararán aquí.
4. Los agentes que han decidido que quieren responder procederán a la creación del nuevo mensaje.
5. Una vez diseñado el mensaje lo registrarán en PESEDIA vía API.

Una vez explicado conceptualmente el funcionamiento de este protocolo de comunicación vamos a justificar por qué se ha decidido este diseño.

¿Cómo mejora nuestra propuesta los mecanismos de comunicación disponibles en PESEDIA actualmente?

Como ya se explicó en el estado del arte, una de las principales razones por las que se decidió coger SPADE como plataforma sobre la que implementar nuestro sistema multiagente era la existencia de experiencia previa en el grupo de investigación responsable del proyecto. En la actualidad, PESEDIA dispone de un sistema multiagente - con un objetivo completamente distinto al que se propone en este proyecto - desarrollado en SPADE que permite analizar el contenido de los mensajes publicados por los usuarios con el objetivo de asesorarles (ver sección 2.1).

A pesar de que los mecanismos disponibles en la actualidad para gestionar la comunicación entre PESEDIA y el sistema multiagente son funcionales, consideramos que no son óptimos y, por tanto mejorables, en base a los siguientes criterios:

- Protocolos empleados actualmente: el sistema actual utiliza dos vías diferentes de comunicación. No obstante, la forma en la que se hace uso de estas vías de comunicación es completamente opuesta a la que se plantea en este trabajo y consideramos que hay margen de mejora.

La primera vía de comunicación, PESEDIA → multiagente, utiliza llamadas HTTPS para indicar al multiagente, a través de un agente maestro con una funcionalidad muy parecida al propuesto, las operaciones que debe realizar. A pesar de ser un protocolo completamente válido, no es el propuesto por SPADE y, por tanto, obliga al desarrollador a aprender y dominar un mecanismo para el que la plataforma, en principio, no está diseñada, añadiendo una capa de complejidad adicional. Además, el uso de este protocolo constriñe la configuración de servidor web y sistema multiagente, limitando las posibilidades de configuración de los mismos a las necesarias para que esta vía de comunicación sea funcional.

La segunda vía de comunicación, multiagente → PESEDIA, utiliza una comunicación basada en ficheros de texto. Una vez que el sistema multiagente completa sus operaciones deposita un fichero de texto que el servidor web está monitorizando. Al igual que en el caso de la vía anterior, esta forma de comunicación es completamente válida pero somete al servidor web a una carga de trabajo que es preferible evitar, ya que la sesión queda bloqueada mientras se monitoriza de manera constante el directorio designado en busca de cambios, impidiendo al usuario el normal uso de la página web hasta que se complete la operación correspondiente.

- Complejidad: al no emplear los protocolos de comunicación propuestos por las diferentes herramientas utilizadas se obliga al programador a intentar razonar, comprender y posteriormente utilizar estos mecanismos de comunicación en lugar de utilizar simplemente los propuestos por las propias herramientas, con una documentación mucho más amplia y consolidada. El objetivo de la nueva propuesta es el de acercar el desarrollo de componentes futuros para la plataforma a los estándares de los diferentes sistemas empleados.
- Propensión a fallos: al utilizar protocolos de comunicación para los que no están pensados los sistemas (como comunicación directa usando llamadas HTTPS o un sistema de comunicación basado en ficheros) aumenta la probabilidad de producirse fallos procedentes del envío de información, como pueden ser fallos en la codificación de los ficheros o en la configuración de los sistemas. Se han tenido previamente muchos problemas con el uso del protocolo SSL (*Secure Sockets Layer*) a la hora de enviar instrucciones desde PESEDIA al sistema multiagente.

Si bien es cierto que el uso de los estándares propuestos por los sistemas no garantiza la desaparición completa de los fallos procedentes de las comunicaciones, podemos afirmar que, gracias a seguir esta buena práctica, será más sencillo detectar, encontrar y corregir estos fallos cuando sucedan.

Por estas razones consideramos que es deseable la modificación de estos mecanismos de comunicación para que se adapten a los estándares con el objetivo de simplificar el proceso de desarrollo y garantizar mayor estabilidad cuando el sistema se encuentre en producción.

Razonamiento para el uso de diferentes protocolos de comunicación con el sistema multiagente

Una duda razonable que le puede surgir al lector es la necesidad de emplear dos protocolos de comunicación diferentes en lugar de ceñirnos únicamente a uno de ellos. A continuación se detallan todas las razones:

- **Reactividad y proactividad de los agentes:** debido a la naturaleza de los agentes necesitamos que los mismos puedan ser capaces tanto de generar estímulos como de reaccionar frente a estímulos externos. En caso de optar por utilizar únicamente la API de Elgg no podríamos obtener agentes reactivos, ya que no habría manera de informar a los agentes de lo que está sucediendo en la red social salvo que ellos necesitasen consultarlo. Este problema podría solucionarse empleando únicamente XMPP, pero optar por esta solución podría generar nuevos problemas, como son la saturación de las vías de comunicación y la sobrecarga de trabajo en el servidor, que se detallan a continuación.
- **Saturación en las comunicaciones y modificación en la estructura de los servidores:** habiendo determinado que, en caso de optar por una vía de comunicación única, XMPP es la única alternativa, surge la necesidad de saber cómo afectará esto a los distintos servidores implicados. En primer lugar sería necesario que el servidor en el que se encuentra alojado PESEDIA estuviese constantemente escuchando posibles llamadas XMPP. Para ello sería necesario modificar el servidor y añadir un socket en el que alojar al cliente XMPP que se encargará de gestionar estas comunicaciones, lo que lo alejaría del diseño original pudiendo conducir a malas prácticas y futuros errores no deseados. Por otro lado el servidor XMPP responsable de gestionar las comunicaciones tendría el doble de carga de trabajo al tener que atender también las peticiones surgidas de los agentes, lo que podría provocar que se viera desbordado y necesitase más recursos hardware de los deseados.
- **Sobrecarga de trabajo en el servidor web:** el servidor web debe encargarse únicamente de servir contenido y debemos buscar constantemente que el desarrollo propio no fuerce una carga de trabajo adicional. Como ya hemos visto en el punto anterior un socket permitiría alojar a un cliente XMPP que atendiese las peticiones de los agentes. No obstante, este socket no sería un web socket al uso ya que, en lugar de estar abierto esperando llamadas HTTP, tendría asociado un agente XMPP que en bucle estaría esperando peticiones XMPP con una conexión persistente, obligando a dedicar recursos constantes para una tarea que puede no ser necesaria. Este uso de recursos penalizaría evidentemente el rendimiento del servidor web, sometido a una gran carga de trabajo de por sí debido a la gran cantidad de eventos que debe gestionar con el normal uso de PESEDIA.

Analizadas estas problemáticas, parece natural querer desacoplar los protocolos, permitiendo que el servidor web atienda y responda llamadas, como es natural, y dividiendo la carga de las comunicaciones en dos alternativas. Por este motivo creemos que la mejor propuesta es utilizar XMPP como mecanismo de transferencia de información en cascada - de la red social hacia abajo - y una API que se encargue de atender las peticiones de los agentes, eliminando así la necesidad de establecer conexiones persistentes entre los servidores y liberando recursos, ya que la comunicación se realizará empleando sesiones únicas.

4.1.3. Esquema de red

Para conseguir que el sistema se acople a los cambios propuestos es necesario modificar la infraestructura de red actual. Aprovechando esta modificación necesaria se van a proponer algunos cambios de manera adicional para intentar dotar a la red de alta disponibilidad y tolerancia a fallos.

En la figura 4.3 podemos encontrar la infraestructura de red actual. Como se puede observar, este esquema de red no se aleja demasiado al modelo de infraestructura clásica

empleada para una aplicación de naturaleza web, a excepción del servidor que alojaría al sistema multiagente. El sistema actual utiliza un sistema de backups tanto locales como almacenados en otro servidor, lo que a pesar de ser útil obliga a detener los servicios y realizar actuaciones manuales en caso de producirse algún tipo de fallo.

En la figura 4.4 podemos encontrar la infraestructura de red propuesta. Esta nueva infraestructura tiene una complejidad superior a la actual debido a la incorporación de mecanismos que garantizarán el funcionamiento de la aplicación a pesar de producirse fallos, replicando los componentes hardware necesarios y mejorando el rendimiento general de la aplicación empleando estrategias de clustering para los servidores web. Aparte de los ya comentados añadidos evidentes que permitan dotar al sistema de alta disponibilidad y tolerancia a fallos, ha sido necesario añadir el nuevo servidor XMPP y las conexiones necesarias para que funcionen los dos protocolos de comunicación. Se ha optado por no replicar el nuevo servidor XMPP y el servidor en el que se alojará el sistema multiagente ya que, en caso de producirse un fallo en uno de ellos, será necesario verificar manualmente las causas. Adicionalmente, al tratarse de un añadido al sistema y no ser un componente que - en caso de fallo - impida el normal funcionamiento de la red, consideramos como opción preferente destinar más recursos al resto de componentes clave del sistema, que sí influyen de manera directa en la experiencia de los usuarios.

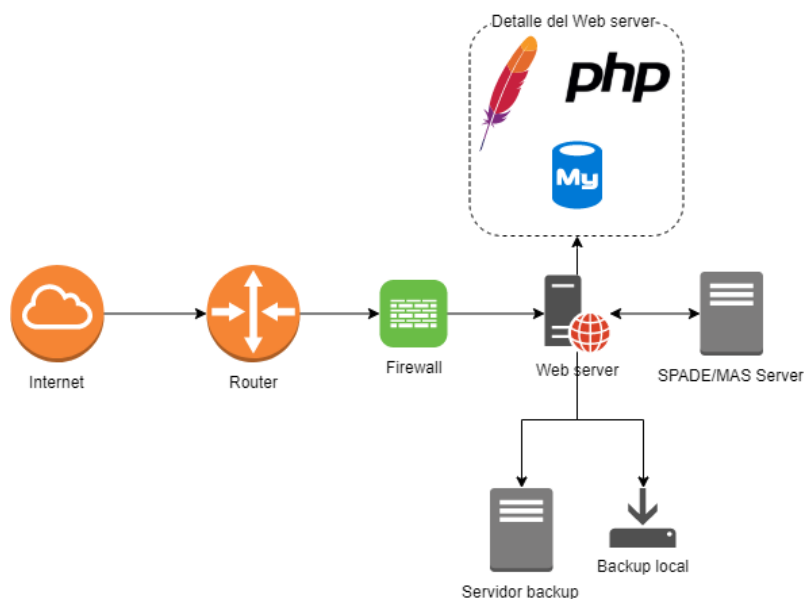


Figura 4.3: Infraestructura de red actual

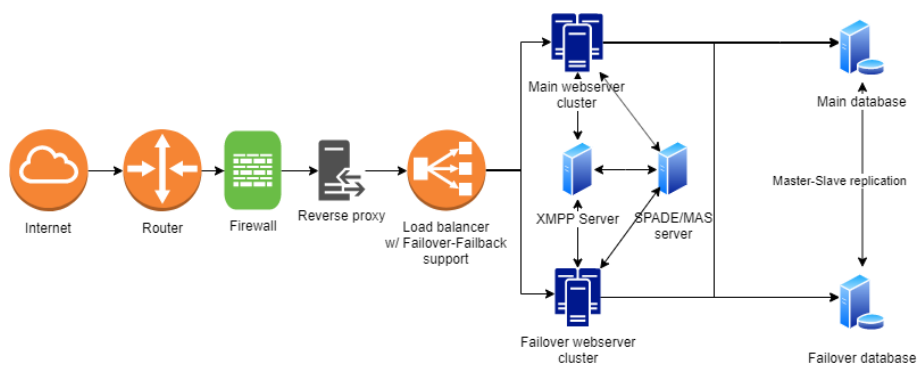


Figura 4.4: Infraestructura de red propuesta

4.2 Especificación formal

4.2.1. Capa de presentación

Aunque la mayor ampliación respecto al trabajo actual consiste en el desarrollo del sistema multiagente y la creación de las funcionalidades necesarias para gestionar las comunicaciones, es necesario ofrecer una interfaz en la propia red social para que el administrador pueda conocer el estado de los agentes y realizar modificaciones en la plataforma a conveniencia. Uno de los requisitos demandaba que la aplicación sea funcional sin importar el tamaño de pantalla. Para satisfacer este requisito, se ha optado por usar un diseño web adaptable (*Responsive design*), que permite modificar la disposición de los elementos en pantalla de manera automática (sin intervención del usuario) según el tamaño del dispositivo, para garantizar la mejor experiencia de uso en todo momento. Con el objetivo de facilitar la visualización de las interfaces conceptuales se ha optado por emplear fotomontajes (mockups) que ilustren el aspecto de las interfaces finales.

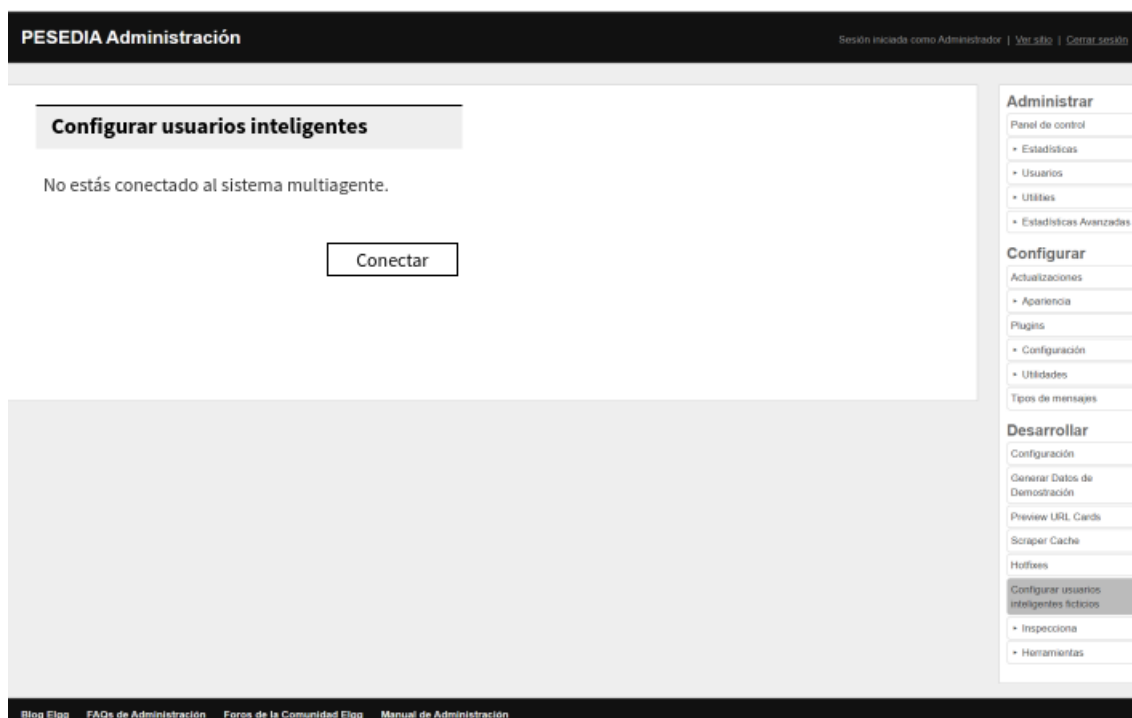


Figura 4.5: Fotomontaje de la versión web para establecer conexión con el sistema multiagente

En la figura 4.5 se muestra la interfaz que podrá ver el administrador tras activar el plugin a través de la opción correspondiente. El único objetivo de esta interfaz es comunicar que el plugin está activado, dando la opción de conectarse al sistema multiagente para continuar con la operativa. Adicionalmente esta es la pantalla que se mostrará cuando el administrador no necesite continuar trabajando con el sistema multiagente siempre que presione el botón desconectar, mostrado en la interfaz de la figura 4.6.

En la figura 4.6 podemos observar la interfaz que se ofrecerá al administrador una vez que realice la conexión con el sistema multiagente. A la izquierda de la imagen se dispondrán las opciones de configuración de los agentes, con diferentes formatos de entrada en base al criterio que se esté modificando. En la parte derecha se mostrará un resumen del estado del sistema a modo de listado, permitiendo así al administrador abstraerse del funcionamiento del sistema multiagente siempre y cuando no tenga que resolver algún error en el mismo. Por último se dispondrá de un botón que permitirá salir de esta vista

títulado "Desconectar" que nos devolverá al estado anterior. Se recuerda que esta desconexión no implica desconectar la web y el sistema multiagente. En caso de desear retirar todos los agentes podrá hacerse tanto desde el menú de configuración como deshabilitando el plugin en el menú correspondiente, siendo esta última opción la única que permite cortar completamente la comunicación entre la web y el sistema multiagente.

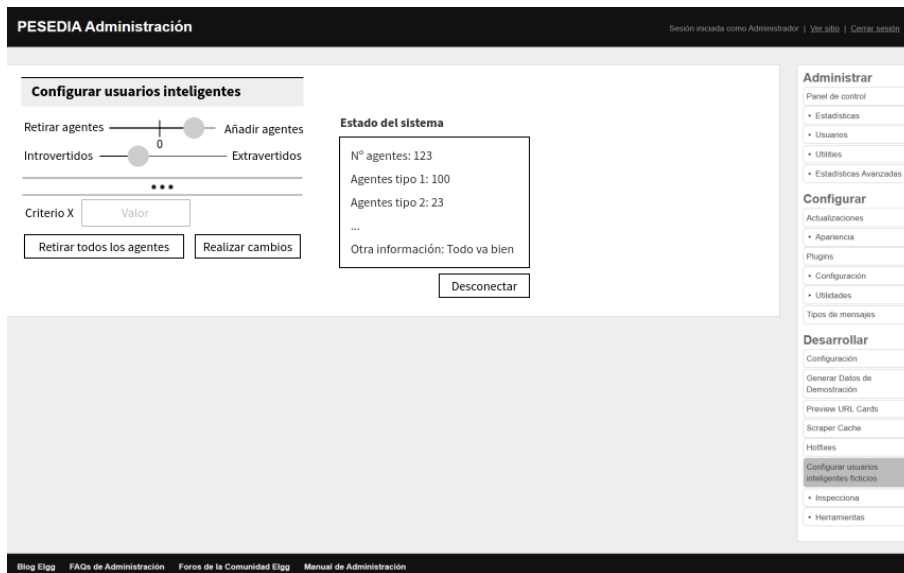


Figura 4.6: Fotomontaje de la versión web para consultar el estado del sistema multiagente y realizar modificaciones

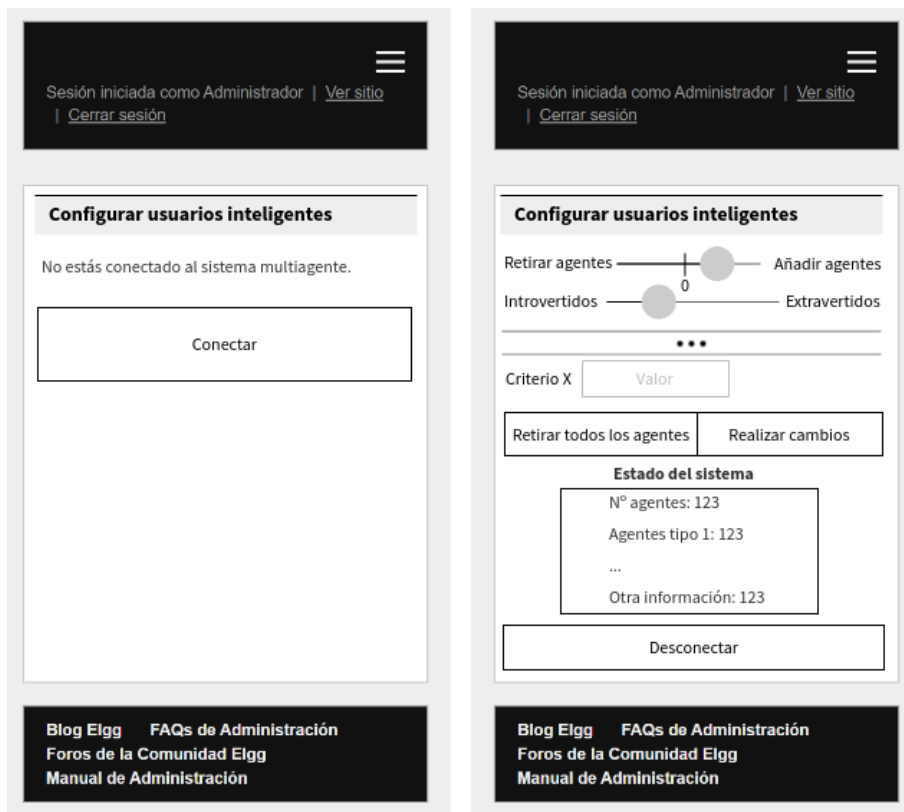


Figura 4.7: Fotomontaje de la versión móvil

Finalmente, en la figura 4.7, pueden observar estas mismas interfaces en su disposición móvil.

Las interfaces mostradas en este punto satisfacen los casos de uso CU01 y CU02. Los casos de uso CU03 y CU04 harán uso de las interfaces proporcionadas por Elgg, por lo que no es necesario diseñarlas. El resto de casos de uso, correspondientes al sistema multiagente, no necesitan hacer uso de interfaces.

4.2.2. Capa de negocio

En este apartado vamos a tratar de representar de manera formal las diferentes actividades que pueden producirse en el sistema. Debido al enorme número de actividades que los agentes pueden llegar a realizar en la red social resulta inviable añadir en este documento todas ellas, ya que excedería en longitud el máximo deseado. Por este motivo hemos optado por priorizar, en primer lugar, la definición de las acciones en las que existe intervención directa por parte de algún usuario - únicamente cuando impliquen una modificación sobre el sistema actual - y en segundo lugar definiremos varias acciones representativas que impliquen la participación y coordinación de los diferentes subsistemas. Debido a la naturaleza autónoma de los agentes se ha optado por emplear diagramas de actividades, ya que es el formato que mejor nos permitirá seguir la secuencia de los sucesos acontecidos en el sistema.

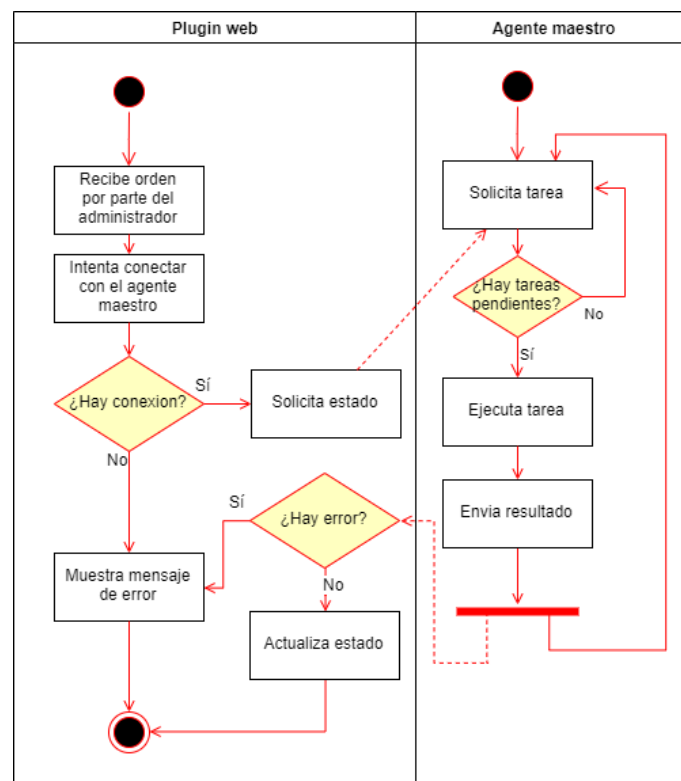


Figura 4.8: Diagrama correspondiente a la acción de solicitud de estado - CU01

La figura 4.8 describe el proceso mediante el cual el plugin web solicita al agente maestro una actualización del estado del sistema. Esta acción puede ocurrir cuando el usuario se conecta al sistema multiagente por primera vez desde la web, en cada ocasión que se refresque la vista correspondiente mientras el administrador no haya desconectado explícitamente (la conexión no es persistente, pero el plugin asume el último estado

fijado) establecida o periódicamente mientras la vista se encuentre activa. Este diagrama cubre el caso de uso CU01.

La figura 4.9 representa los pasos a seguir por el sistema cuando el administrador desee añadir o retirar agentes de la plataforma, tal y como se solicitaba en el caso de uso CU02. Este diagrama es también aplicable en el caso de desactivación del plugin, puesto que en ese caso se solicitará la eliminación de todos los agentes. Adicionalmente este diagrama también representa el caso de uso CU05 ya que también detalla el proceso automático que seguirá el agente maestro para gestionar los agentes.

Para satisfacer los casos de uso CU03 y CU04 vamos a representar en el diagrama de actividad la operativa que seguirá un agente para responder un mensaje del usuario. Para ello partiremos de un estado inicial en el que se asume que el usuario ha introducido un mensaje. El final de este proceso ocurre cuando el plugin web almacena el mensaje en la base de datos de PESEDIA, ya que el resto del proceso de visualización no forma parte de las modificaciones asociadas a este proyecto. De esta manera queda cubierta la interacción con los agentes al entablar una conversación con ellos (CU04) y la persistencia de mensajes para su posterior visualización (CU03). En la figura 4.10 puede observarse este proceso de manera gráfica.

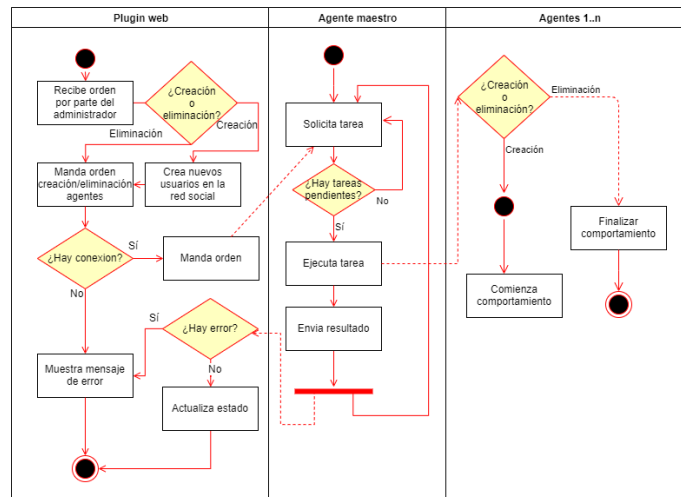


Figura 4.9: Diagrama correspondiente a la acción de añadir o retirar agentes de la plataforma - CU02

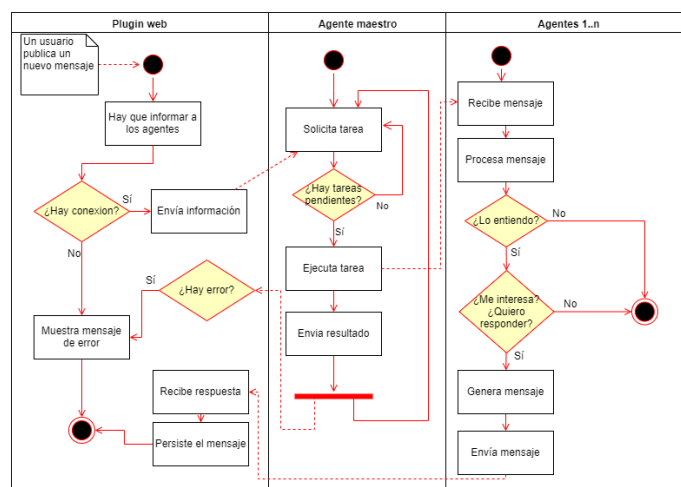


Figura 4.10: Diagrama correspondiente a la acción de responder y persistir un mensaje por parte de los agentes para su posterior visualización - CU03 y CU04

Todas las acciones definidas hasta ahora son las que afectan de manera directa a la interacción con el usuario, pero existen multitud de acciones más que podrían realizar los agentes: añadir a un amigo, dar "me gusta" a un comentario, unirse a un grupo, etc. La cantidad de acciones es tal que necesitaríamos un proyecto al completo para analizar todas en detalle. Por este motivo vamos a simplificar este apartado y representaremos únicamente la acción de añadir un amigo, ya que exige proactividad a los agentes y ninguna de las anteriores cubre ese hecho, tal y como se puede ver en la figura 4.11.

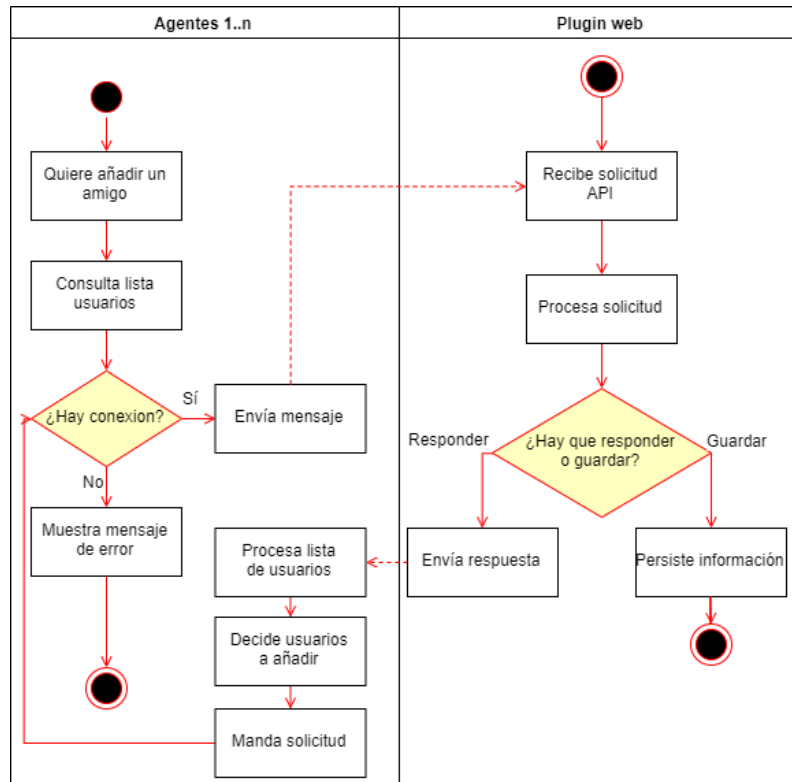


Figura 4.11: Diagrama correspondiente a la acción de añadir un amigo por parte de los agentes - CU06

4.2.3. Capa de persistencia

La capa de persistencia es la que vamos a analizar en menor detalle debido a que las ampliaciones que vamos a realizar harán uso, en la mayoría de casos, de los mecanismos de persistencia ya disponibles. Como hemos comentado previamente, Elgg proporciona una base de datos basada en MySQL, con su propia estructura, a la que nosotros accedemos mediante las funciones que facilita el propio framework. Estas funciones hacen el papel de un pseudo-ORM (*Object-Relational mapping*) por lo que no es necesario conocer la estructura interna de la base de datos, sino conocer las diferentes entidades con las que podremos trabajar. Como esta estructura ya viene dada y no es propia de este proyecto consideramos que no aporta un valor añadido explicar en estas páginas cómo gestiona Elgg la información. No obstante en caso de querer entrar en detalle se recomienda consultar el trabajo de José Alemany Bordera "PESEDIA. Red social para concienciar en privacidad" [2] en el que se trata el tema, así como la propia documentación de la plataforma ¹.

En cuanto a los agentes, al tratarse en esta aproximación inicial de elementos volátiles, se crearán y destruirán sin posibilidad de ser reutilizados en el futuro. Por tanto, la

¹<http://learn.elgg.org/es/stable/design/database.html>

información que les pudiese resultar de interés (código de usuario asignado, modelo de usuario, lista de amigos, etc.) se almacenará en la memoria del propio agente, ya que la cantidad de información a almacenar es baja y no consideramos justificable la inversión de esfuerzo en el diseño de una base de datos, o un sistema de ficheros, para almacenar datos que, por el momento, van a ser de tipo clave-valor y no van a tener relación entre ellos. Una de las propuestas de trabajos futuros (que se tratará en sucesivos capítulos) consistirá en dotar de "memoria" a los agentes, por lo que será necesario replantear y expandir este apartado.

4.3 Modelos de los agentes

4.3.1. Factores

Con el objetivo de modelar los diferentes agentes que podrán coexistir en el sistema es necesario describir una serie de características (o factores) de manera que, a partir de su agrupación, obtengamos unos diseños que representen de manera fidedigna los distintos perfiles de usuario más habituales en las redes sociales. En la tabla 4.1 se agrupan estos factores junto a todos sus posibles valores.

Factor	Valor						
Edad	Joven	Adulto	Senior				
Género	Hombre	Mujer					
Tiempo de uso	Usual	Ocasional	Eventual				
Número de amigos	Bajo	Medio	Alto				
Personalidad	Reservado	Egocéntrico	Promedio	Modelo a seguir			
Patrones de uso	Introvertido	Novedoso	Versátil	Comunicador			
Perfil de privacidad	Minimalista	Auto censor	Optimizador	Equilibrado	Distribuidor selectivo	Maximizador	
Experiencia privacidad	Novato	Casi novato	Algo novato	Algo experto	Casi experto	Experto	

Tabla 4.1: Factores que definen los modelos de los agentes, con sus posibles valores

A continuación se detallan y explican los diferentes factores seleccionados tras la realización del estado del arte, vista en el capítulo 2:

- **Edad:** en todos los trabajos revisados la edad es uno de los factores recurrentes. Habitualmente, estos estudios categorizan a los usuarios en tres grandes grupos. El primero de ellos corresponde a los jóvenes, que son usuarios entre 14 y 24 años. El segundo grupo es el de usuarios adultos, entre 25 y 64 años. El último grupo es el de usuarios senior, que tienen 65 años o más. Este factor será utilizado, junto al género, para definir los diferentes grandes tipos en los que agruparemos nuestros modelos.
- **Género:** las propuestas estudiadas agrupan a los usuarios en dos géneros: hombre y mujer, por lo que nos ceñiremos a ambos. Este factor será utilizando, junto a la edad, para definir los diferentes grandes tipos en los que agruparemos nuestros modelos.
- **Tiempo de uso:** este factor está incluido en el trabajo de Alarcón et al. [1], en el que se agrupan usuarios en base a sus patrones de uso. Hemos decidido extraerlo de la clasificación por grupos que se hace en el trabajo citado ya que es un factor que nos permite determinar la frecuencia con la que los agentes realizarán acciones sin ceñirnos estrictamente a su perfil de patrones de uso. Los valores que puede tomar este factor son: usual (el usuario hace uso de la red social múltiples veces al

día), ocasional (el usuario hace uso de la red social diariamente o cada dos días) y eventual (el usuario accede a la red social una/dos veces a la semana, o menos).

- Número de amigos: segundo factor incluido en el trabajo de Alarcón et al. [1]. Hemos decidido extraerlo de la clasificación por grupos que se hace en el trabajo citado ya que es un factor que nos permite hacer un cruce con los modelos propuestos en el resto de trabajos seleccionados. Los valores que puede tomar este factor son: bajo (menos de 50 amigos), medio (entre 50 y 100 amigos) y alto (más de 100 amigos).
- Patrones de uso: este factor utiliza los grandes grupos propuestos por Alarcón et al. [1] para clasificar usuarios según sus patrones de uso. Los valores que puede tomar el factor corresponden con cuatro de los diferentes grupos expuestos en el trabajo. A pesar de usarlos como referencia, no vamos a ceñir el comportamiento de los agentes a lo estrictamente definido en dicho trabajo, ya que tendremos en cuenta el resto de comportamientos descritos en los otros trabajos referenciados en el apartado 2.4.3.
- Personalidad: en este factor se representan los cuatro perfiles de personalidad propuestos en el trabajo de Gerlach et al. [23].
- Perfil de privacidad/Experiencia privacidad: los dos últimos factores representan los modelos propuestos en el trabajo de Wisniewski et al. [51]. A la hora de utilizar estos factores, cada modelo podrá tener varias configuraciones asignadas a pares que se asignarán de manera aleatoria, ya que no se ha encontrado una relación estricta entre los perfiles de personalidad y sus preferencias y experiencia en gestión de privacidad.

A modo de resumen en referencia a la asignación de los valores, tal y como se ha comentado, los modelos se organizarán por tipos según sus valores para los factores género y edad. Para cada tipo se dispondrán diferentes modelos en base al resto de factores a excepción de los perfiles de privacidad y experiencia, que podrán ser varios por cada modelo y se elegirán aleatoriamente entre las opciones para cada uno de los agentes.

4.3.2. Acciones

Cada modelo podrá realizar una serie de acciones en base a su perfil de usuario. Para cada acción se espera que el agente tenga un interés base mayor o menor en realizarla, dependiendo igualmente de su perfil.

Como se puede observar en la tabla 4.2 la única acción que los agentes deberán cumplir obligatoriamente es la de añadir un amigo, aunque su interés en hacerlo sea bajo. Hemos decidido esto ya que, en caso de permitir que los agentes no tengan interés en realizar ninguna acción, podríamos llegar a crear agentes completamente pasivos.

En el caso de tratarse de un usuario real, este comportamiento puede tener sentido ya que representaría a un usuario que se registra en la red social por curiosidad, pero no realiza acciones en la misma. Desde el punto de vista computacional, esto no tiene sentido ya que implicaría una inversión de recursos que no se aprovechará y, en caso de querer usuarios que hagan nada, es mejor crear un perfil de usuario sin necesidad de acompañarlo de un agente que no tenga comportamiento.

Aparte de los motivos ya mencionados, hemos decidido escoger la acción de añadir a un amigo ya que es, posiblemente, la acción más simple que se puede realizar dentro de una red social y la que más interés suele generar en una persona ya que, habitualmente, cuando un usuario se registra en una red social lo hace con la intención de mantener

contacto con sus amigos y es habitual tener usuarios con un volumen importante de amigos que no participan activamente en la red social.

El último factor, "preferencia privacidad", indica la preferencia que tendrá el usuario en relación a los diferentes aspectos de privacidad y alcance de la información, de manera representativa y no funcional. Este campo será más útil para, a simple vista, comprender las diferencias entre los distintos modelos durante la lectura del documento, que para validar su funcionamiento una vez estén los agentes en funcionamiento, ya que su toma de decisiones en aspectos de privacidad quedará determinada por los factores "perfil de privacidad" y "experiencia privacidad" definidos en el apartado anterior.

Factor	Valor			
Añadir amigos	Bajo	Medio	Alto	
Actualizar su estado	Ninguno	Bajo	Medio	Alto
Dar likes	Ninguno	Bajo	Medio	Alto
Generar comentarios	Ninguno	Bajo	Medio	Alto
Interés en grupos	Ninguno	Bajo	Medio	Alto
Redifundir comentarios	Ninguno	Bajo	Medio	Alto
Rellenar su perfil	Ninguno	Bajo	Medio	Alto
Etiquetar fotos	Ninguno	Bajo	Medio	Alto
Subir fotos	Ninguno	Bajo	Medio	Alto
Crear álbumes	Ninguno	Bajo	Medio	Alto
Preferencia privacidad	Más privado	Más público		

Tabla 4.2: Factores que definen las acciones que podrán realizar los agentes, con sus posibles valores según el interés mostrado

4.3.3. Modelos de ejemplo

Con el objetivo de representar los diferentes perfiles de usuario que pueden encontrarse en una red social se han diseñado una serie de modelos basados en los factores y las acciones definidas en esta sección. En las tablas 4.3 y 4.4 se adjuntan los dos modelos propuestos para su implementación en este trabajo ya que resulten muy representativos y muestran grandes diferencias en su comportamiento, para que así sea más sencillo comprobar y contrastar las diferencias entre ambos, verificando el correcto funcionamiento del sistema.

Tipo 1: Hombre joven - "Poco activo"

Factor	Valor					Factor	Valor
Edad	Joven					Añadir amigos	Bajo
Género	Hombre					Actualizar su estado	Bajo
Tiempo de uso	Eventual					Dar likes	Bajo
Número de amigos	Bajo					Generar comentarios	Bajo
Personalidad	Promedio					Interés en grupos	Bajo
Patrones de uso	Novedoso					Redifundir comentarios	Ninguno
Perfil de privacidad	Equilibrado	Minimalista	Optimizador	Equilibrado	Auto censor	Rellenar su perfil	Bajo
Experiencia privacidad	Casi novato	Casi novato	Casi novato	Casi novato	Casi novato	Etiquetar fotos	Ninguno
						Subir fotos	Bajo
						Crear álbumes	Ninguno
						Preferencia privacidad	Más privado

Tabla 4.3: Perfiles de usuario y actividad Tipo 1

Este primer modelo representa a un perfil de usuario correspondiente a un varón con una edad comprendida entre los 18 y los 24 años, poco activo en la red social y que muestra, por tanto, poco interés en las diferentes acciones disponibles en la plataforma.

Tipo 2: Mujer adulta - "Muy activa/Influencer"

El segundo modelo representa a una usuaria de la red social muy activa y que trata de sacar el mayor provecho de las diferentes funciones ofrecidas por la plataforma con el objetivo de conseguir atraer a más seguidores.

Factor	Valor	Factor	Valor
Edad	Adulto	Añadir amigos	Alto
Género	Mujer	Actualizar su estado	Alto
Tiempo de uso	Usual	Dar likes	Alto
Número de amigos	Alto	Generar comentarios	Alto
Personalidad	Promedio	Interés en grupos	Medio
Patrones de uso	Comunicador	Redifundir comentarios	Alto
Perfil de privacidad	Equilibrado	Rellenar su perfil	Alto
Experiencia privacidad	Casi novato	Etiquetar fotos	Alto
	Minimalista	Subir fotos	Alto
	Optimizador	Crear álbumes	Alto
	Equilibrado	Preferencia privacidad	Más público
	Auto censor		
	Casi novato		
	Casi novato		
	Casi novato		
	Casi novato		
	Casi novato		

Tabla 4.4: Perfiles de usuario y actividad Tipo 2

4.4 Plan de trabajo

Para la consecución de este trabajo, debido a la complejidad y extensión del mismo, se va a seguir un plan basado en hitos, en los que se tratará de dotar a la herramienta de nuevas funcionalidades de manera iterativa. Para este plan de trabajo se va a considerar únicamente el proceso de desarrollo y pruebas, ya que el proceso de implantación no dependerá del equipo de trabajo.

1. **Desarrollo de los mecanismos de conexión entre PESEDIA y el sistema multi-agente:** el primer hito de este proyecto consistirá en el desarrollo de las dos vías de comunicación explicadas previamente. Este proceso de comunicación se repetirá posteriormente (con las modificaciones necesarias) en el desarrollo de cada una de las acciones. Para la ejecución de este hito se plantean los siguientes objetivos:
 - a) **Creación del plugin en PESEDIA:** en primer lugar será necesario desarrollar un nuevo plugin básico en PESEDIA.
 - b) **Activación de la API en PESEDIA y desarrollo de end-point para consulta de estado:** una vez creado el nuevo plugin, la primera ampliación que se realizará será la creación de una API con una ruta a la que se podrá consultar desde el exterior para saber si la red social está disponible. Debe tenerse en cuenta que, para el consumo de la API, debe ser necesario disponer de un mecanismo de autenticación basado en claves.
 - c) **Dotación de funcionalidad XMPP a PESEDIA:** Elgg por defecto no dispone de ninguna librería que permita comunicarse a través del protocolo XMPP. Por tanto será necesario encontrar - o diseñar - una librería que permita llevar a cabo este tipo de comunicación.
 - d) **Desarrollo básico del agente maestro:** la primera iteración del agente maestro necesitará ofrecer a PESEDIA una forma de consultar si el SMA está disponible (de manera análoga a la función previamente explicada en la API) y, adicionalmente, deberá poder consumir la función de consulta de estado disponible en la API de PESEDIA.
 - e) **Consulta del estado del SMA desde PESEDIA:** hay ciertas funcionalidades que requerirán a PESEDIA conocer si el sistema multiagente está disponible.

La primera de ellas será la activación del plugin, por lo que será necesario modificar el proceso de activación para que realice una consulta al agente maestro y, en caso de no obtener respuesta, impida habilitar el plugin.

2. **Creación de nuevos agentes:** una vez desarrollada la funcionalidad previa y validada la posibilidad de enviar información entre ambos sistemas, podemos comenzar el desarrollo de funcionalidades en el sistema multiagente. Para ello se plantean los siguientes objetivos:
 - a) **Modelado de los agentes:** el primer paso requerirá el desarrollo de los modelos propuestos para nuestros agentes, por lo que, aparte de crear las clases necesarias, será necesario que el agente maestro disponga de la funcionalidad necesaria para identificar el tipo de cada agente que se encuentre en el sistema.
 - b) **Proceso de creación de nuevos agentes en el SMA:** una vez implementados los modelos de los agentes, tendremos que permitir al agente maestro que cree nuevos agentes bajo demanda, teniendo en cuenta el número y tipo de los mismos.
 - c) **Creación de nuevos usuarios en PESEDIA:** de manera análoga, será necesario en PESEDIA desarrollar un mecanismo de creación de usuarios aleatorios teniendo en cuenta el tipo (en base a los propuestos) y el número. Una vez creados los usuarios debe informarse al sistema multiagente.
 - d) **Modificación del agente maestro para que reciba la petición de PESEDIA y pueda responder con el resultado:** una vez que el agente maestro sea capaz de recibir de PESEDIA las instrucciones para los agentes se tendrá que ampliar el desarrollo previo para que el agente maestro, en primer lugar, proporcione a los agentes toda la información que les atañe dentro de la red social y, en segundo lugar, responda a PESEDIA una vez creados los agentes.
3. **Primera acción: Añadir amigos.** La primera acción a desarrollar será la posibilidad de añadir amigos en la red social. Para lograrlo se deben cumplir los siguientes objetivos:
 - a) **Agentes amigos en el SMA:** el primer paso consiste en dotar a los agentes de la capacidad de simular círculos de amigos dentro del propio sistema multiagente para replicar todas las acciones que, entre seres humanos, tendrían lugar fuera de la red social.
 - b) **Traspaso de información a la red social:** simultáneamente, en la red social deben desarrollarse los mecanismos necesarios para que los agentes puedan enviar a PESEDIA solicitudes de amistad.
 - c) **Envío de información al sistema multiagente:** PESEDIA debe informar al sistema multiagente de todos los usuarios que vayan añadiéndose al sistema multiagente para que los agentes determinen si quieren ser amigos suyos. Adicionalmente debe enviar al sistema multiagente de todas las peticiones de amistad que vayan recibiendo, para que los agentes decidan si quieren aceptarlas.
 - d) **Agentes amigos en PESEDIA:** es necesario que los agentes puedan conocer el estado de su lista de amigos en la red social, por lo que deben tener en cuenta la información del mismo a la hora de tomar decisiones.
4. **Otras acciones:** debido a la complejidad del proyecto consideramos que los pasos previos permitirían asentar la piedra angular sobre la que continuar el desarrollo y ejemplificar los puntos claves descritos en este trabajo, por lo que no se detallarán

acciones adicionales. A modo de planificación se considerará este paso como un periodo de tiempo indefinido que terminará una vez el responsable del proyecto quede satisfecho con todas las funcionalidades.

En la figura 4.12 se encuentra un diagrama de Gantt en el que se muestra el plan de trabajo definido.



Figura 4.12: Plan de trabajo usando un diagrama de Gantt

4.5 Presupuesto

A pesar de que el proyecto resultado de este trabajo ha sido realizado en solitario (y ha sido necesario realizar más labores además de las descritas en el plan de trabajo), en este apartado se va a realizar un cálculo basándonos en el plan de trabajo teórico descrito en el apartado anterior, para el que se necesitaría un número de recursos superior al utilizado. A continuación se detallan los mismos, junto al coste aproximado que supondrían.

Para el cálculo de los salarios se han utilizado los valores medios definidos en la página de búsqueda de empleo indeed.es² a fecha de escritura del trabajo. Para los materiales se ha hecho una estimación en base a los precios encontrados en diferentes tiendas web.

Recursos humanos

- **Jefe de proyecto** (1): responsable de coordinar el trabajo de los diferentes miembros del equipo de trabajo. Salario: 35.673 euros anuales.
- **Investigador** (1): responsable de llevar a cabo todas las iniciativas relacionadas con la innovación: estudio y selección de propuestas relevantes, modelado de los agentes, definición de acciones, probabilidades, etc. Salario: 30.432 euros anuales.

²<https://www.indeed.es/salaries>

- **Analista (1):** responsable de coordinar el trabajo de los programadores, de la validación y calidad del software producido y del mantenimiento de la base de código. Salario: 29.212 euros anuales.
- **Programador senior (2):** responsables de la implementación, uno del apartado web y otro del sistema multiagente. 26.452 euros anuales
- **Administrador de sistemas (1):** Responsable del mantenimiento de la infraestructura y los diferentes recursos hardware necesarios. Salario: 23.838 euros anuales.

Recursos materiales

- **Ordenadores personales (6):** Equipos básicos para labores de desarrollo, ofimática, etc. Coste estimado: 900 euros (usando como referencia los modelos de la serie Lenovo Thinkpad)
- **Servidor de desarrollo (2):** Equipos sobre los que se desplegarán las instancias de desarrollo del proyecto y la base de código. Coste estimado: 600 euros (usando como referencia los modelos más económico de la serie Dell PowerEdge)
- **Infraestructura:** materiales necesarios para interconectar los diferentes componentes hardware y la conexión a Internet. Se asume que, al tratarse de un proyecto para la universidad, se utilizaría la red vigente.
- **Otros: material de oficina, eventualidades:** se asume que, al tratarse de un proyecto para la universidad, los costes asociados correrían a cargo de la misma, empleando los recursos existentes.

4.6 Conclusiones

En este capítulo se han visto las diferentes consideraciones a seguir durante el desarrollo de la aplicación, detallando en cada punto las motivaciones que inducen a realizar las modificaciones necesarias en el sistema actual, así como las diferentes ampliaciones sobre el mismo. Se han detallado los principales flujos de trabajo para conseguir que los agentes puedan interactuar con la red social, así como los esbozos de las nuevas interfaces que podrán utilizar los administradores de PESEDIA.

Adicionalmente se han analizado los factores y acciones que se utilizarán en los agentes para caracterizar los diferentes tipos de usuarios y sus comportamientos en base a los trabajos analizados en el estado del arte. A continuación se han detallado los modelos de ejemplo que se van a desarrollar en este trabajo.

Finalmente, se ha detallado tanto el plan de trabajo a seguir como el presupuesto necesario previsto en base a todos los factores analizados.

CAPÍTULO 5

Implementación y validación

En este capítulo va a explicarse el proceso de implementación y la validación de los nuevos elementos desarrollados. Con el objetivo de facilitar la organización y lectura del documento se va a seguir una estructura similar a la propuesta en el plan de trabajo descrito en el capítulo anterior.

5.1 Implementación

5.1.1. Hito 0 - Preparación de la infraestructura

Aunque en el plan de trabajo se asumía la existencia de la infraestructura, ha sido necesario desplegar una nueva infraestructura ya que la actual no cumplía los requisitos necesarios para el desarrollo. En el capítulo anterior ya se comentó esta mejora en la infraestructura y las posibles modificaciones para ofrecer alta disponibilidad. Finalmente se ha optado por emplear el sistema de contenedores Docker para el despliegue de esta infraestructura por las siguientes razones:

- La primera razón ha sido la necesidad inicial de crear una infraestructura de desarrollo. Aunque el GTI-IA posee una instancia de desarrollo sobre la que realizar pruebas, el flujo de trabajo actual exigía tener una instalación local para realizar la labor de desarrollo, lo que resultaba poco práctico debido a la complejidad de la instalación.
- Utilizar tecnología de contenedores permitirá que, en el futuro, sea sencillo utilizar diferentes herramientas de orquestación para gestionar el proceso de desarrollo y despliegue de la aplicación, así como el uso de herramientas como Kubernetes¹ que permitan desplegar los diferentes servicios en un sistema basado en cloud, pudiendo conseguir de esta manera la alta disponibilidad sin necesidad de utilizar multitud de máquinas físicas.

La estructura de contenedores implementada finalmente utiliza como referencia la propuesta basada en servidores con alta disponibilidad, encontrada en la subsección 4.1.3. Para ello se disponen diferentes servicios que, gracias a su interconexión, permiten satisfacer las necesidades de ambos subsistemas. A continuación se detalla esta infraestructura, condensada en las figuras 5.1 y 5.2:

¹<https://kubernetes.io/>

1. **Base de datos:** el primer contenedor contiene la infraestructura de la base de datos utilizando la imagen base "mysql:5.7". Este contenedor expone el puerto 3306 para que se pueda acceder a la base de datos, tanto desde el resto de contenedores, como desde programas que se estén ejecutando en la máquina host. Este contenedor puede comunicarse con el contenedor 2 (servidor web) y con el contenedor 3 (PHPMyAdmin). Este contenedor necesita hacer uso de volúmenes locales en los que guardar la información de la base de datos, consiguiendo así mantenerla entre diferentes sesiones.
2. **Servidor web:** el segundo contenedor contiene la infraestructura correspondiente al servidor web en el que se alojará la red social. Este contenedor utiliza la imagen "php:7-apache" con un gran número de modificaciones para incluir todas las extensiones necesarias para que Elgg pueda funcionar. Este contenedor expone los puertos 80 y 443 para poder servir la página a la máquina local. Puede comunicarse con el contenedor 1 (base de datos). Este servidor hace uso de volúmenes locales para sincronizar el contenido de la página web.
3. **PHPMyAdminServer:** el tercer contenedor no es necesario para el funcionamiento de la plataforma, pero proporciona acceso a la herramienta PHPMyAdmin que era utilizada por algunos de los desarrolladores del grupo de investigación para consultar el contenido de la base de datos, por lo que se ha decidido mantener en esta infraestructura. Este contenedor utiliza la imagen "phpmyadmin/phpmyadmin", expone el puerto 80 y únicamente puede conectarse con el contenedor 1 (base de datos).
4. **Multiagente:** el cuarto y último contenedor es el responsable de gestionar el servidor XMPP sobre el que se harán las comunicaciones y de ejecutar el sistema multiagente. Para conseguirlo partimos de la imagen `unclev/prosody-docker-extended:0.10` a la que se ha añadido el software necesario para poder ejecutar Python (necesitado por SPADE) y un servidor de base de datos PostgreSQL, necesitado por Prosody para almacenar la información de usuarios y contraseñas, salas de conversación, grupos, historial de chats, etc. Gracias a SPADE podremos aprovechar toda esa estructura de información sin necesidad de implementar código adicional, ya que la plataforma ofrece la funcionalidad necesaria para poder aprovecharla. Este contenedor hace uso de volúmenes locales para, en primer lugar, almacenar la información de la base de datos y, en segundo lugar, sincronizar el código del sistema multiagente.

Como se puede deducir, los componentes principales de esta infraestructura son los contenedores 2 y 4, ya que son los que albergan la base de código que se va a desarrollar. Gracias a esta infraestructura se garantiza disponer de todos los elementos necesarios para la ejecución del proyecto de manera sencilla y sin necesitar entrar a la configuración particular de varios servidores web. A pesar de ello, vamos a necesitar desarrollar una base de código que permita la comunicación entre los dos sistemas y es lo que se tratará de conseguir en el siguiente hito.

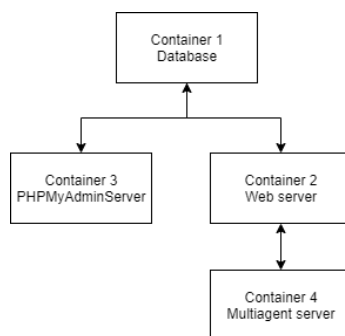


Figura 5.1: Esquema de comunicación de los contenedores en Docker

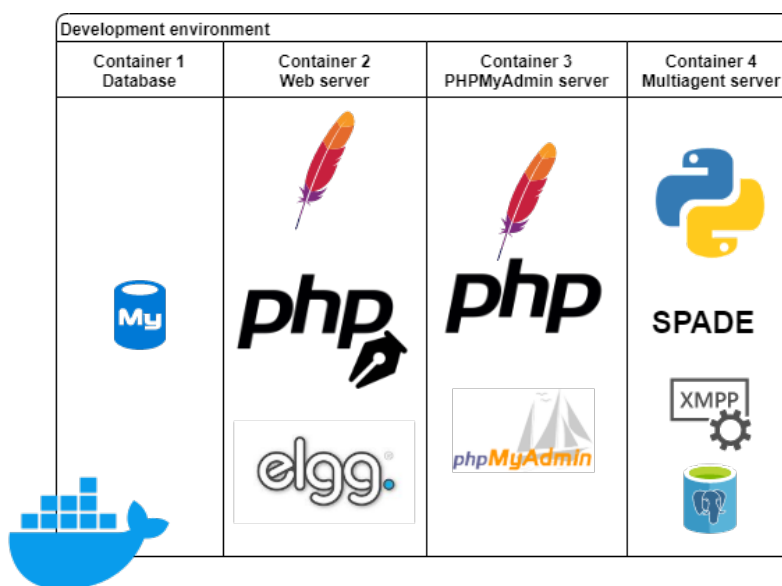


Figura 5.2: Infraestructura Docker utilizada en el desarrollo del proyecto

5.1.2. Hito 1 - Mecanismos de comunicación

Como se vio en el capítulo anterior, este proyecto va a hacer uso de dos vías de comunicación para enviar información entre los dos principales sistemas. Este proceso no es trivial ya que ninguna de las plataformas dispone por sí misma de las herramientas necesarias, por lo que es necesario el desarrollo de un conjunto de mensajes propios (utilizando JSON) a modo de protocolo interpretable por las dos plataformas. Para facilitar la explicación de este proceso, se van a detallar los pasos que se han seguido para conseguir que las plataformas puedan consultar si el otro servicio está disponible a través de un sistema de handshake (saludo).

Para llevar a cabo la explicación de este proceso van a comentarse en primer lugar las modificaciones realizadas en el sistema multiagente y a continuación las realizadas en el servidor web.

- Modificaciones en SPADE:** partiendo de una instalación básica de SPADE, la primera labor a realizar fue la creación del llamado “agente maestro” que realizará la función de interfaz con el exterior, exponiendo la funcionalidad de SPADE a otros sistemas. El proceso de creación de agentes se detalla en la documentación de la plataforma, por lo que se emplaza al lector a su consulta.

La manera en la que SPADE determina qué acciones deben realizar los agente se basa en un sistema de "comportamientos". Existen comportamientos de diferentes tipos: cíclicos (que se ejecutan de manera constante), periódicos (que se ejecutan según cierta frecuencia), basados en temporizadores, etc.

El primer comportamiento a implementar será una función cíclica que permita al agente maestro leer y procesar mensajes de manera constante. El objetivo de esta función será recibir un mensaje de texto en formato JSON, procesar el mensaje, extraer qué operación desea ejecutarse, ejecutar dicha operación y, finalmente, enviar el resultado de la llamada.

Antes de proceder a implementar la función de handshake, es necesario determinar qué tipo de entradas podrá recibir el agente maestro. En una situación ideal la llamada será correcta, tanto en forma (es un JSON válido) como en contenido (la operación está soportada), pero es posible que se incumpla una de esas dos condiciones.

Es por ello que, la primera tarea a realizar, es la definición de los JSON que se van a enviar a través de los sistemas. Para el caso de las operaciones de entrada, el input definido esperamos que tenga un contenido similar al siguiente:

```
{
  "operation": nombre_operacion,
  "additional_parameters": [array de parámetros, si los hubiese]
}
```

Este tipo de entrada nos permite tanto definir la operación a realizar como señalar una serie de parámetros a procesar por la función. En el caso de la función de handshake no es necesario indicar parámetros opcionales, por lo que bastaría con indicar la operación, que en este caso se ha decidido denominar como "mas_operation_handshake".

De cara a enviar la respuesta, como ya hemos mencionado existen varias posibilidades. La primera de ellas es que todo haya funcionado correctamente, por lo que se mandará un JSON con un contenido similar al siguiente:

```
{
  "mas_response": "ok",
  "input": JSON de entrada,
  "result": resultado si lo hubiese
}
```

Y, en caso de error, un JSON similar al siguiente:

```
{
  "mas_response": "error",
  "input": JSON de entrada,
  "cause": motivo del error }
}
```


Este formato permite que el receptor sepa el resultado de la llamada, pudiendo relacionarlo con la función que lo ha ejecutado y además conocer, en caso de que la llamada haya sido correcta, el resultado o, en caso de que la llamada haya fallado, el motivo del error.

Con todo lo descrito anteriormente, queda resuelta en el multiagente la respuesta a las peticiones de estado por parte del servidor web. El siguiente paso es permitir al sistema multiagente consultar el estado del servidor web. Para conseguirlo se ha implementado otro comportamiento, en este caso periódico con una frecuencia de 30 segundos, cuya función es consultar el estado del servidor web para que, en caso de no obtener una respuesta de manera consecutiva tres veces (aproximadamente un minuto y medio en total) detenga todos los agentes del sistema.

Para hacer esta petición se ha hecho uso del paquete "response" no incluido por defecto en SPADE, y se ha hecho una llamada al endpoint correspondiente en la API PESEDIA utilizando la clave generada en el servidor web.

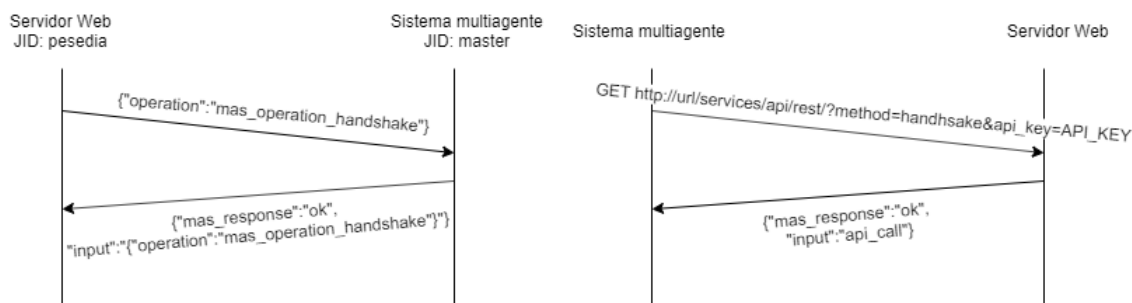


Figura 5.3: Esquema de llamadas para responder a peticiones de estado desde la red social y el sistema multiagente

- Modificaciones en Elgg:** para poder comenzar este proceso es necesario crear una extensión de Elgg ya que, acorde a las guías de estilo de esta plataforma, no es recomendable modificar el núcleo de la aplicación de manera directa. El proceso de creación de plugins se explica en el trabajo de fin de máster de José Alemany Bordera [2], por lo que se recomienda su consulta en caso de querer obtener más información acerca de este tema.

La primera funcionalidad a desarrollar en el plugin es una petición de handshake al sistema multiagente durante la activación del mismo. Para lograrlo es necesario utilizar el sistema de mensajería XMPP, no disponible por defecto en el servidor. Para lograr esta funcionalidad se ha hecho uso de la librería "xmpp-php"², que incorpora los métodos necesarios para poder gestionar el envío y la recepción de mensajes utilizando este protocolo.

Una vez instalada la librería, en el evento responsable de la activación del plugin se añade la comprobación del estado del multiagente enviando al agente maestro un JSON siguiendo la estructura vista anteriormente. En caso de no recibir una respuesta afirmativa a lo largo de una serie de iteraciones se asumirá que el sistema multiagente no está disponible y se impedirá la activación del plugin.

Para poder hacer uso de la API de Elgg y programar la función que permita a los agentes conectarse desde el sistema multiagente es necesario habilitar dos módulos.

²<https://github.com/Norgul/xmpp-php>

El primero de ellos habilita la API de Elgg y el segundo permite generar claves de consulta a la API (mencionados en otras secciones de este documento). Por tanto, es necesario añadir como requisitos del nuevo plugin dichas extensiones. Una vez activados los plugins necesarios, generada la clave API y proporcionada al multi-agente, crearemos un nuevo endpoint que, al ser consultado, responderá al usuario con un JSON afirmativo, idéntico a los anteriores.

En la imagen 5.3 se definen los diagramas de comunicación esperados en caso de poder realizar estas conexiones de manera satisfactoria.

5.1.3. Hito 2 - Creación de nuevos agentes y usuarios

Una vez generados los mecanismos de comunicación entre la red social y el sistema multiagente, el siguiente paso a desarrollar es la capacidad de añadir nuevos agentes, con sus correspondientes usuarios en la red social. Para ello es nuevamente necesario realizar modificaciones en los dos sistemas. Como el proceso de comunicación será idéntico al explicado en el punto anterior, cambiando el contenido de los mensajes, cuando se vaya a tratar el envío de mensajes se mencionará únicamente el contenido de los JSON (en el caso de tratarse de comunicaciones XMPP), para evitar ser reiterativos.

- **Modificaciones en SPADE:** para poder crear los agentes, SPADE necesitará recibir de Elgg una entrada con los parámetros que definan el comportamiento que deberá tener la nueva acción "mas_operation_add_agents". Esta acción necesitará saber el tipo de agentes que se desean añadir y el número de los mismos, la lista de identificadores que se les han asignado y los nombres.

Cuando el agente maestro reciba la petición JSON (vía XMPP) la analizará y procederá a la creación de los nuevos agentes, gracias a que SPADE permite crear y ejecutar nuevos agentes desde el comportamiento de un agente en funcionamiento.

Para la creación de estos nuevos agentes se creará una clase base, llamada "*BaseUser*" que contendrá toda la información y comportamiento de carácter genérico - independiente del tipo - y adicionalmente se crearán dos nuevas clases "*OpenUser*" y "*ShyUser*", que representarán los dos tipos de usuarios propuestos en el capítulo anterior.

El agente maestro creará de manera iterativa los nuevos agentes, guardándose sus identificadores en una lista para poder posteriormente enviarles información y, a la vez que los crea, se encarga de comunicar a todos los agentes del sistema la creación de los nuevos agentes.

Estos nuevos agentes serán los responsables de simular el comportamiento de los usuarios. El primer comportamiento del que dispondrán consistirá en un método que se ejecutará únicamente una vez y cuya única función será la de imprimir por pantalla un mensaje que muestre el resultado de la creación.

Una vez terminado este proceso de creación, el agente maestro tendrá que mandar la respuesta a la red social, para que el administrador sepa si se han creado los agentes, o si ha habido algún fallo en el proceso. En caso de éxito se mandará un mensaje de éxito similar al visto en el hito anterior. En caso de fallo, se añade una nueva consideración a las ya vistas en el anterior hito, que es la posibilidad de que

el mensaje JSON que se envía incluya (por fallos de programación) tipos de usuarios no contemplados en el sistema multiagente. Por ello durante la creación, en caso de detectar este tipo de error, se procederá a la eliminación de todos los agentes creados a partir de esta última solicitud y se informará al administrador a través de un mensaje de error que le aparecerá en la vista correspondiente del plugin web.

- **Modificaciones en Elgg:** continuando con la ampliación del plugin, será necesario crear una ventana desde la que el administrador pueda configurar el número de agentes que quiere añadir tanto a la red social como al sistema multiagente en base a los dos tipos de agentes posibles.

Para el desarrollo de este apartado, se ha utilizado como ejemplo un plugin llamado *"hypeFaker"*³ que permite generar contenido ficticio en la red social con el objetivo de poder hacer pruebas en la misma. Una de las funcionalidades de dicho plugin permite la creación de usuarios, por lo que se ha trasladado esa funcionalidad a nuestro nuevo plugin.

Una vez solucionado el problema de creación de usuarios, se requiere enviar al sistema multiagente la información necesaria para que pueda, tanto crear los agentes, como proporcionarles la información básica que necesitarán para poder comunicarse con la red social: su identificador de usuario en PESEDIA y su nombre de usuario asignado. Esta información se extraerá de la red social a través de una consulta a su base de datos justo después de registrar los nuevos usuarios ficticios. A continuación se adjunta un JSON que ejemplifica el contenido que PESEDIA enviará al sistema multiagente.

```
{
  "operation": "mas_operation_add_agents",
  "additional_parameters": {
    "types": [
      {"type": "AGENT_TYPES_1", "count": 1},
      {"type": "AGENT_TYPES_2", "count": 1}
    ],
    "pesedia_ids": [1, 2],
    "user_names": ["John Doe", "Jane Doe"]
  }
}
```

5.1.4. Hito 3 - Creación de círculos de amigos dentro del sistema multiagente

La primera acción compleja que deben realizar los agentes es la de añadir a otros usuarios como amigos. Una persona puede tener amigos en su vida personal que no sean amigos en sus redes sociales y viceversa. Esta dualidad es importante ya que obliga a considerar dos escenarios muy distintos de cara a la implementación. Debido a limitaciones temporales, en esta primera versión del programa se ha decidido incluir únicamente la acción que permita a los agentes hacerse amigos de otros - representando a los amigos "reales" - dentro del sistema multiagente.

Se ha tomado esta decisión ya que en los anteriores hitos ya se han llevado a cabo las implementaciones necesarias para ejemplificar los pasos que se llevarían a cabo para

³<https://github.com/hypeJunction/hypeFaker>

representar esta acción: lectura de información en la red social, envío vía XMPP, respuesta vía API. Por este mismo motivo se ha decidido representar únicamente el proceso de comunicación dentro del sistema multiagente, ya que era un caso no cubierto en el resto de hitos que, además, compartirá gran similitud con el procedimiento no realizado de traspaso de conocimiento a PESEDIA, ya que la toma de decisiones a la hora de añadir o aceptar un amigo será idéntica.

Es por este motivo que, para este hito, solamente ha sido necesario realizar modificaciones en el sistema multiagente.

- **Modificaciones en SPADE:** para poder llevar a cabo este hito es necesario modificar ampliamente el agente base. En primer lugar será necesario definir dos nuevos comportamientos que permitan tanto tomar la iniciativa a la hora de añadir otro agente como amigo, como tomar la decisión de aceptar la petición que provenga de otro agente.

Previamente al desarrollo de estos dos comportamientos, será necesario definir una función que permita al agente decidir si quiere, o no, añadir a un amigo. Esta función, a la que se ha llamado "get_friend_desire_mas", utilizará los diferentes factores relevantes, junto a la intención de realizar una actividad que tienen los agentes según su tipo, y obtendrá un valor numérico que represente el deseo actual que tiene el agente de añadir un amigo.

La intención de añadir un amigo varía en el rango [0-100] %. El porcentaje de probabilidad viene definido por los siguientes factores, y es representada por la siguiente función:

$$deseo = \max(fA * 25 + fP * 8,33 - (nA * \text{neg}(1 - nA * 0,3) + vR), fP + 1 + vR)$$

donde:

- **deseo:** representa el resultado de la operación.
- **fA:** representa el deseo innato de realizar la acción que tiene el agente en base a la probabilidad de realizar acciones definida en el diseño del modelo de agente. Este valor puede ser bajo (valor 1), medio (valor 2) o alto (valor 3). Este factor es el que mayor influencia positiva tiene en el cálculo, pudiendo llegar a generar un deseo con valores (25, 50, 75) % respectivamente.
- **fP:** representa el factor de personalidad del agente. Si bien el primer factor (fA) ya tenía en cuenta la personalidad a la hora de determinar la probabilidad de que el agente quiera realizar la acción, pues el modelo del agente viene determinado por su personalidad, es necesario tenerlo en cuenta nuevamente con un factor de impacto inferior para representar la varianza que puede imponer el tipo de personalidad. En un futuro este factor podría ser modificado por el factor emocional, en el que se trate de representar el sentir del agente en un momento determinado concreto, mucho más apropiado para evitar la reutilización del factor de personalidad, aportando más riqueza al comportamiento del agente. Este factor fP tiene una influencia notable, aunque menor que el factor fA, a la hora de calcular el deseo del agente, pudiendo proporcionar un valor entre (8.33, 16.66, 24.99).
- **nA:** representa el número de amigos que el agente tiene en la actualidad. Como hemos visto en los trabajos analizados en el estado del arte, los usuarios tienden a tener un número de amigos relativamente homogéneo, relacionado directamente con su personalidad y patrones de uso. Por tanto, es natural

buscar que los agentes tengan menos intención de añadir nuevos amigos según vaya creciendo su número. Se puede considerar un factor de alto impacto ya que por sí mismo puede llegar a eliminar el deseo del agente de buscar o aceptar nuevos amigos.

- **vR**: el comportamiento de los humanos no siempre es consecuente, por lo que se ha añadido un factor de aleatoriedad que servirá como estímulo o como detrimento a la hora del cálculo final. Este valor variará, de manera aleatoria, en el rango $[-1, 1]$, por lo que su impacto es mínimo.

La función matemática propuesta genera un descenso lineal en el deseo que tendrán los agentes de añadir amigos, que será proporcional al número de amigos que haya podido conseguir el agente, sin llegar a eliminar la posibilidad de que el agente intente añadir un nuevo amigo, para así garantizar su autonomía. Aunque resulta evidente, el valor máximo que puede alcanzar el deseo del agente es del 100 % que se daría en el caso de un agente extremadamente social, y ese valor disminuiría linealmente según el agente crease su círculo de amigos. Esta función permite representar fielmente la diferencia entre el número de amigos que tiene un agente poco social y uno muy social, llegando a observar valores medios muy similares a los encontrados en los trabajos analizados. Gracias a la fórmula propuesta, cobran sentido los dos comportamientos que tendrá el agente a la hora de añadir o aceptar un amigo, ya que en ambos casos se hará uso de ella.

Al tratar de añadir otro agente como amigo, lo primero que hará el agente interesado es comprobar si tiene el deseo de hacerlo y para ello hará uso de la siguiente fórmula:

$$intencion = deseo > rand(1, 100)$$

En caso de cumplirse, procederá a consultar los agentes disponibles en la red, utilizando para ello la funcionalidad de "Presencias" que ofrece SPADE. Gracias a esa funcionalidad podemos obtener los diferentes clientes XMPP que están conectados al servidor de chat y nos permite saber si el cliente actual está suscrito (es amigo) de otro cliente. Como es obvio, esta funcionalidad nos permite representar a la perfección el comportamiento que buscamos ya que, en caso de ser amigos, el gestor de SPADE nos dirá que ambos agentes se encuentran suscritos entre sí y, en caso contrario, consideraremos que no son amigos. Por tanto, en caso de querer añadir un amigo, los agentes seleccionarán aleatoriamente uno de los disponibles, que no sea ya su amigo (estado de relación "none" en SPADE), y le mandará la solicitud.

El segundo comportamiento, aprovechando la posibilidad que nos ofrece SPADE para sobrescribir el comportamiento que tienen los agentes al enviar y recibir peticiones de suscripción, nos permitirá definir cómo queremos que actúen. Cuando estén enviando la solicitud de suscripción los agentes continuarán realizando otras acciones, pero esperarán asíncronamente a recibir una respuesta. Esto es posible ya que SPADE incluye estados intermedios que consideran que un agente ha mandado una solicitud a otro, utilizando un estado denominado "ask".

En el momento en que el agente receptor comience a procesar la solicitud, decidirá si quiere o no tener un nuevo amigo haciendo uso de las dos formulas anteriores. En caso afirmativo procederá a aceptarla, informando al agente emisor y mandándole a la vez una petición de suscripción. En caso contrario la rechazará e informará al emisor. Cuando el emisor reciba la respuesta, en caso de haber satisfecho su petición, aceptará la solicitud de suscripción del receptor y finalizará el proceso, quedando ambos suscritos entre sí y pasando el estado en SPADE a "both". En caso de

rechazo, el emisor tendrá que eliminar su estado de solicitud, pasando nuevamente el estado en SPADE a "none" y añadiendo al agente receptor a una lista en la que estarán todos los agentes que han ido rechazando sus solicitudes de amistad, pudiendo de esta manera evitar que un agente intente insistentemente añadir a otro y representando así de una manera más correcta el comportamiento habitual en una red social, ya que el escenario más repetido es que al ser rechazado, el interesado no trate de volver a añadir a la otra parte, aunque es posible que posteriormente la otra parte quiera añadir por su cuenta al interesado, pudiendo éste decidir si quiere aceptar su solicitud o no.

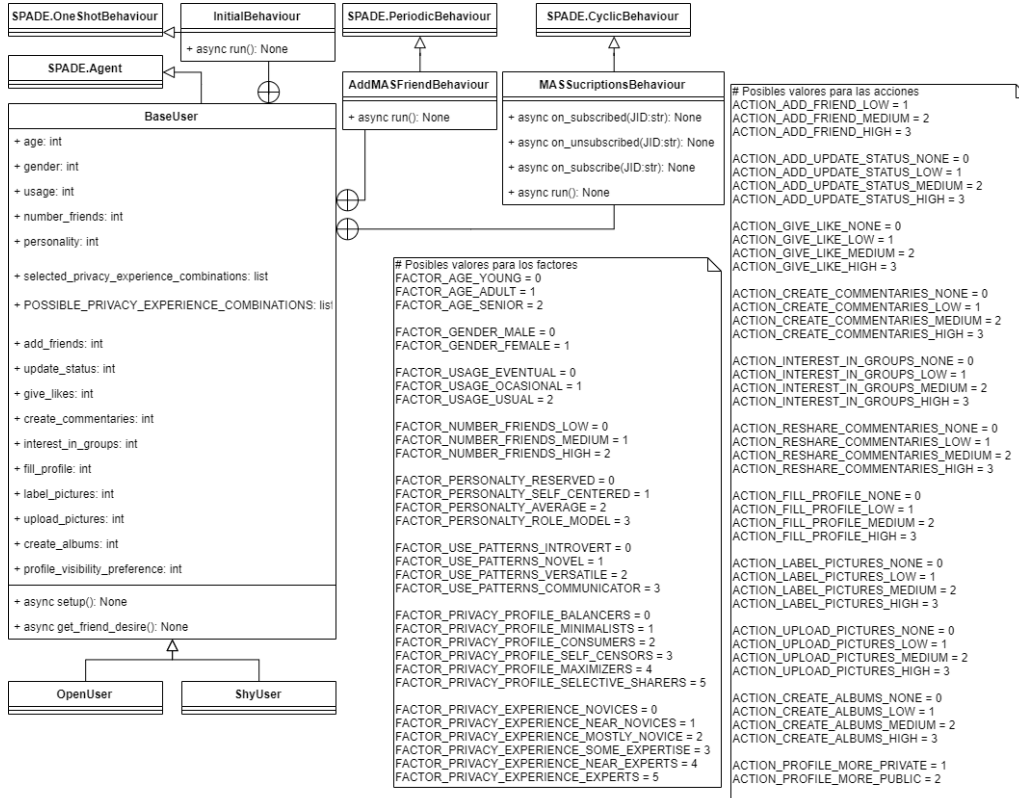


Figura 5.4: Diagrama de clases de tipos de usuarios y sus comportamientos

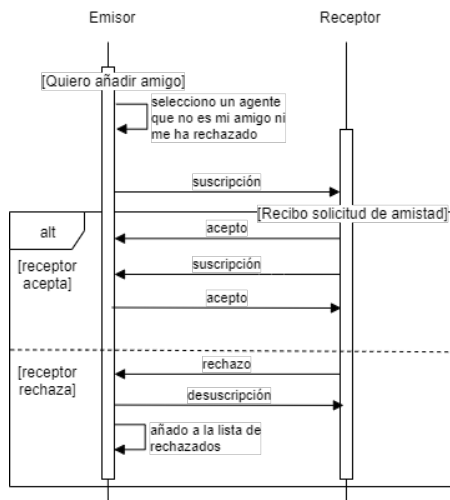


Figura 5.5: Diagrama de secuencia mostrando el proceso de comunicación entre agentes al añadir un amigo

Con las modificaciones anteriormente expuestas se considera resuelto este hito. En la figura 5.4 se encuentra el diagrama de clases final, resultado de este proyecto. En la figura 5.5 se define gráficamente el comportamiento que tendrán los agentes, descrito previamente, a la hora de enviar y recibir peticiones de amistad.

De este modo concluye la fase de implementación del proyecto y comienza en la próxima sección el proceso de validación y análisis de resultados generados a partir del trabajo realizado dentro de este apartado.

5.2 Validaciones

5.2.1. Validación del Hito 1

Para la validación de este hito es necesario comprobar qué sucede cuando una de las vías de comunicación no esté disponible. Para ello se probará en primer lugar la comunicación entre PESEDIA y el sistema multiagente utilizando XMPP y, a continuación, la comunicación entre ambos utilizando la API de Elgg.

En primer lugar accederemos a la sección de plugins dentro del panel de administración de PESEDIA. Una vez dentro, podemos comprobar que aparezca el nuevo plugin desarrollado. Como se puede ver en la figura 5.6, no podemos activar el plugin porque faltan dependencias. Si hacemos click en el nombre del plugin podremos ver una ventana en la que se ofrece la información con los requisitos para poder activar el plugin. En la figura 5.7 podemos ver esta ventana, junto a la relación de plugins necesarios.

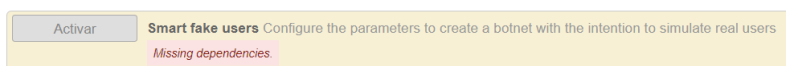


Figura 5.6: No se puede activar el plugin porque faltan dependencias

Smart fake users
Configure the parameters to create a botnet with the intention to simulate real users

Información | Archivos | Dependencias

Tipo	Nombre	Valor de Test	Valor Actual	Comentario
Requiere	Versión Elgg	>= 2.3	2.3.14	OK
Requiere	Plugin: web_services	Cualquiera	1.9	OK
Requiere	Plugin: api_admin	Cualquiera	1.0	OK
Provee	Plugin: smart_fake_users	1.0.0	--	OK

Figura 5.7: Información de dependencias del plugin

Una vez activados los dos plugins requeridos, al recargar la ventana se habilitará el botón de activación de nuestro plugin. En caso de no haber conectado el sistema multiagente, cuando el administrador intente activar el plugin aparecerá una ventana de error, mostrada en la figura 5.8, en la que se indica que no se obtiene respuesta del multiagente y, aunque se puede tratar de un problema de red, se recomienda al usuario comprobar si el sistema multiagente está activo.

Error Fatal

COMUNICACIÓN CON EL MAS - No se ha obtenido respuesta, comprobar que el MAS está en funcionamiento

Log at time 1574639311 may have more data.

Figura 5.8: Error al activar el plugin cuando el sistema multiagente no está disponible

Una vez habilitado el sistema multiagente, o solventados los problemas de red, se mostrará al usuario la ventana principal (ver figura 5.9) donde podrá comenzar a configurar el multiagente.

Desarrolladores : Configurar usuarios inteligentes ficticios

Generar usuarios

Estado del sistema

No hay agentes en el sistema.

Usuarios tímidos (Tipo 1)

Usuarios abiertos (Tipo 2)

Figura 5.9: Ventana de configuración del plugin

Falta por validar el método en la API de Elgg que nos permita comprobar desde el multiagente si la red social está disponible. Para conseguirlo, se ha implementado un método en el agente maestro que, al ser invocado, hará esta llamada y nos devolverá el resultado. En las figuras 5.10 y 5.11 se muestra el resultado de estas llamadas, utilizando el cliente Postman⁴.

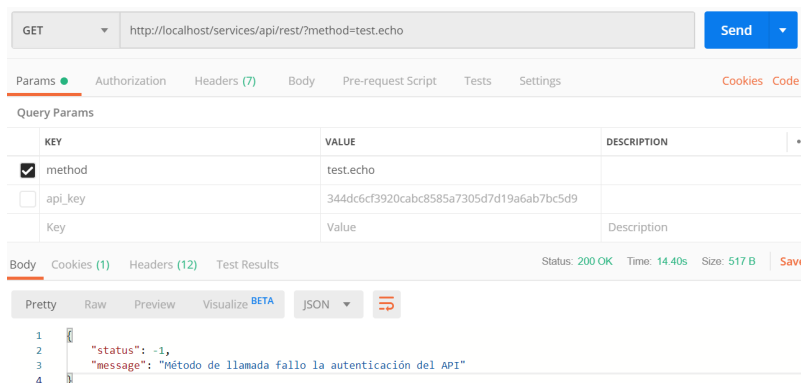


Figura 5.10: Mensaje de error que muestra la API cuando no se utiliza la clave necesaria para hacer consultas

⁴<https://www.getpostman.com/>

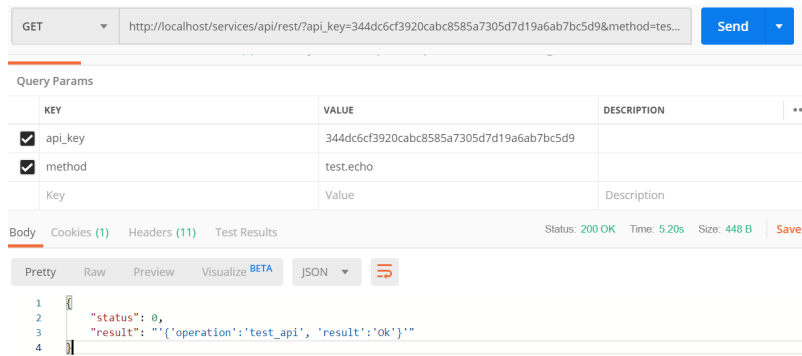


Figura 5.11: Mensaje que recibirá el sistema multiagente cuando PESEDIA funcione correctamente

5.2.2. Validación del Hito 2

Una vez garantizada la conexión entre servicios, vamos a proceder con la validación de creación de usuarios y agentes. Utilizando el panel de administración, vamos a crear 10 usuarios de cada tipo. Una vez termine el proceso, podemos comprobar que en PESEDIA se han creado correctamente los usuarios visitando la ventana de "Miembros", tal y como se muestra en la figura 5.12.

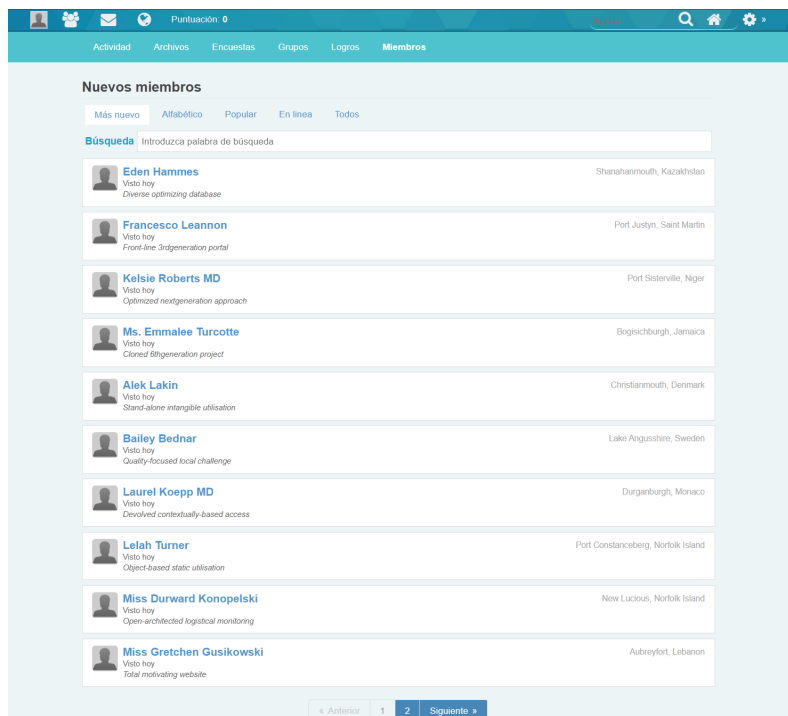


Figura 5.12: Usuarios ficticios creados en PESEDIA

Para validar la creación de los usuarios en el sistema multiagente, es necesario ejecutarlo en modo depuración y comprobar que, tras la finalización del método, se encuentran todos en memoria de manera correcta, tal y como se muestra en la figura 5.13.

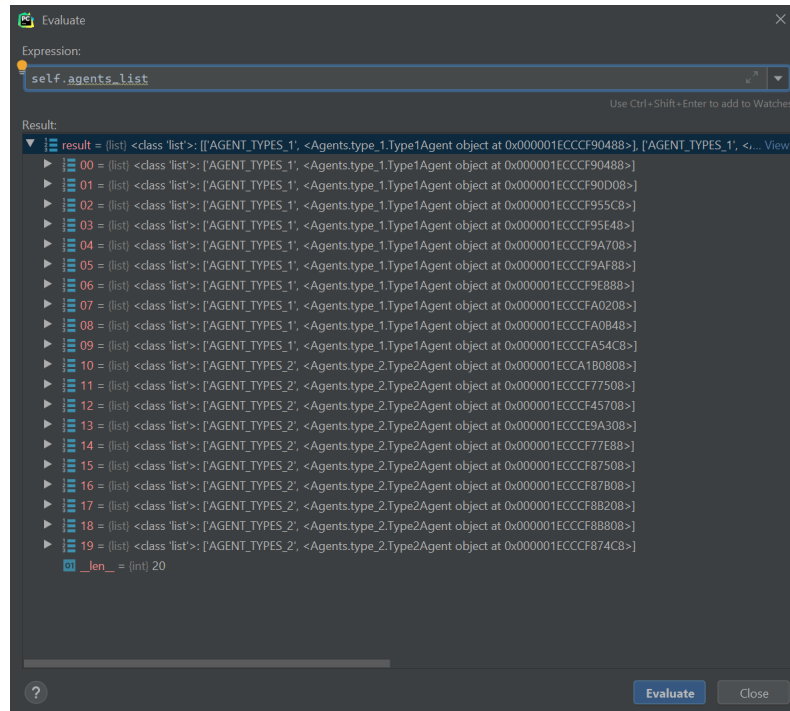


Figura 5.13: Ventana de depuración del sistema multiagente en el que se puede ver la lista de agentes creados en el sistema

Si el administrador consulta ahora en PESEDIA, podrá ver que la ventana de configuración del plugin ha cambiado ligeramente y, en la parte derecha, donde antes se le indicaba que no había agentes ahora le aparece una lista con el tipo y número, como se muestra en la figura 5.14

Desarrolladores : Configurar usuarios inteligentes ficticios

Generar usuarios

Usuarios tímidos (Tipo 1)

Usuarios abiertos (Tipo 2)

Estado del sistema

Usuarios tímidos (Tipo 1): 10

Usuarios abiertos (Tipo 2): 10

Figura 5.14: Ventana de configuración del plugin con información de agentes

5.2.3. Validación del Hito 3

Para validar que los agentes forman círculos de amigos dentro de la red social, es necesario validar diferentes conceptos.

La primera validación a realizar consiste en la validación de la evolución en el deseo que tendrán los agentes por añadir amigos. Para poder validar esta función, se ha modificado ligeramente el comportamiento definido previamente en el que los agentes actúan tras determinar que quieren añadir a un amigo. La modificación realizada consiste en "engañar" al agente para que piense que cada vez que envía una solicitud de amistad siempre es aceptada. Para este cálculo se ha utilizado la media de los resultados obtenidos con una muestra de 100 agentes de cada tipo.

De esta manera, conseguimos obtener unos resultados, mostrados en la figura 5.15, que se aproximan en gran medida a los esperados. Por un lado, se puede observar que los agentes de tipo 1 (tímidos) tienen un deseo inicial inferior a los agentes de tipo 2 (abiertos) y, de igual manera, su deseo por añadir amigos decrece y desaparece mucho más rápido. En la práctica, el deseo de los agentes no llegará nunca a cero ya que se forzaba un valor mínimo por código, pero en esta validación ha decidido no incluirse ya que no es necesario de cara a validar el decremento, que es el ítem que se buscaba validar.

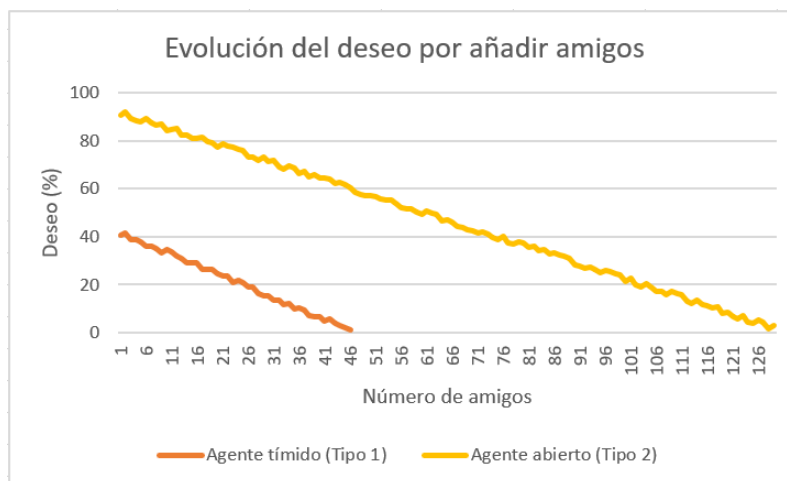


Figura 5.15: Evolución del deseo por añadir amigos que tendrán los agentes

Una vez validado que el decremento en el deseo por realizar la acción funciona de la manera esperada, es necesario remarcar que a partir de ahora serán los agentes receptores los responsables de responder a las solicitudes y, además, añaden los agentes a la lista de ignorados en caso de ser rechazados, revirtiendo el cambio introducido en la anterior validación.

Para comprobar este comportamiento se ha hecho uso de una muestra de 100 agentes, dividida en un factor de 50-50 entre ambos tipos. El comportamiento esperado en esta prueba es que todos (o casi todos) los agentes de tipo abierto se hagan amigos entre sí y puedan llegar a hacerse amigos de aproximadamente la mitad de los agentes tímidos. A su vez, los agentes tímidos deberían recibir solicitudes de todos los agentes abiertos por lo que, en caso de aceptar a aproximadamente la mitad, es poco probable que puedan llegar a hacerse amigos de otros agentes tímidos.

En caso de no haber optado por añadir la restricción de que un agente no pueda solicitar amistad múltiples veces a un agente que le ha rechazado, se daría el caso de que todos los agentes terminarían antes o después haciéndose amigos de todos.

En la figura 5.16 se muestra el resultado de esta validación. Como se puede observar los agentes de tipo abierto son, en el peor de los casos, amigos de al menos 50 agentes, por lo que se cumple el primer supuesto. Adicionalmente se observa que, de mediana, consiguen hacerse amigos de menos agentes de los esperados. Esto puede deberse a que los agentes tímidos se han hecho amigos entre sí más de lo previsto, bajando su deseo antes de que los agentes abiertos les hayan podido preguntar. Esta suposición parece confirmarse al observar en los agentes tímidos que, si bien de mediana parecen tener el número de amigos muy cercano al esperado, supera ampliamente en uno de los cuartiles este valor.

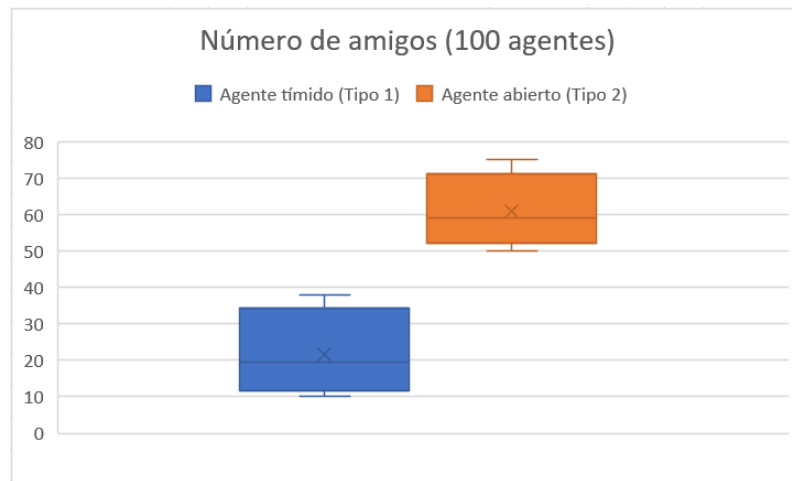


Figura 5.16: Número de amigos que tienen los agentes al finalizar la ejecución, usando un reparto de 50-50 según su tipología

Para tratar de obtener una representación gráfica de lo sucedido se ha realizado una prueba adicional, con un número de agentes y un reparto diferente al utilizado en la anterior prueba. Los resultados de esta prueba se encuentran en la figura 5.17. Como se puede observar, una vez más los agentes abiertos son amigos entre sí y tienen un número de amigos superior a los agentes tímidos. Por tanto, se puede concluir que el diseño de la red de amigos que se generará dependerá, en una medida importante, a las decisiones que tomen los agentes tímidos.

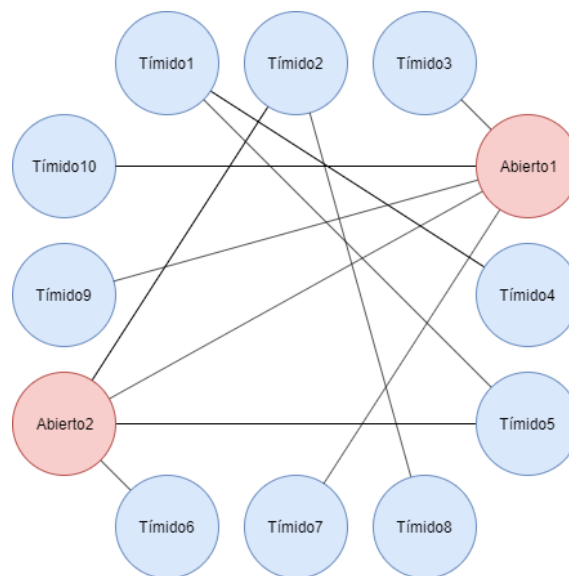


Figura 5.17: Red de amigos generada tras una ejecución con 12 agentes

5.2.4. Pruebas de rendimiento

Para medir el impacto que tiene el sistema multiagente se ha hecho uso de la herramienta *"memory-profile"*⁵ para comprobar el consumo de memoria del mismo. Gracias a este análisis se han extraído las siguientes conclusiones:

⁵<https://pypi.org/project/memory-profiler/>

1. El agente maestro, debido a tener un mayor número de comportamientos que los otros tipos de agentes, tiene un consumo de memoria de 27.4MB aproximadamente, mientras que en los agentes de tipo 1 y 2 este consumo es cercano a los 12.3MB.
2. El servidor de Prosody sobre el que se envían las comunicaciones tiene un impacto casi inapreciable en el sistema. Únicamente puede llegar a producir conflicto cuando está transfiriendo mensajes. El impacto es insignificante en comparación a los propios agentes.
3. La decisión en los estilos de comportamiento para cada acción que puedan tener los agentes tiene un impacto mayor del esperado, debido a la arquitectura de SPA-DE. En el futuro es recomendable revisar y modificar los tipos de comportamiento seleccionados, si es posible, para intentar minimizar el número de recursos necesarios.

En la figura 5.18, puede verse una gráfica que ilustra el consumo de memoria producido por una ejecución con 100 agentes, aunque el consumo de memoria depende en gran medida de los comportamientos que están ejecutando los agentes en el momento de la medición, por lo que puede haber una ligera desviación en los resultados obtenidos.

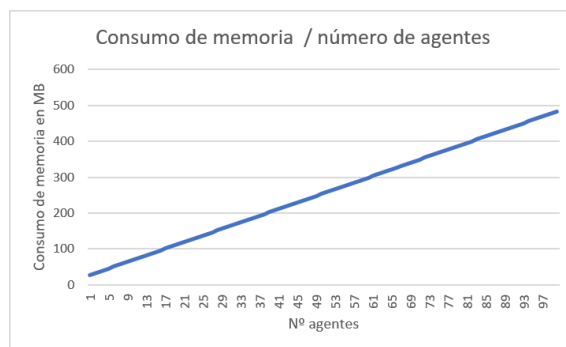


Figura 5.18: Análisis del impacto en memoria del sistema multiagente

5.3 Conclusiones

En este capítulo se han detallado los pasos que se han llevado a cabo para lograr completar con éxito el desarrollo del producto propuesto en este proyecto. Finalmente se ha realizado una batería de pruebas, funcionales y de rendimiento, para comprobar y asegurar el correcto funcionamiento del mismo.

Aunque finalmente no se ha podido implementar toda la funcionalidad propuesta inicialmente, como se puede observar en este apartado se han cubierto de manera satisfactoria los diferentes puntos clave del proyecto, ejemplificando con éxito la comunicación entre sistemas, la creación de agentes de diferentes tipologías y un ejemplo de su comportamiento esperado, por lo que en el futuro podrá continuarse el desarrollo utilizando la implementación actual como base sobre la que seguir expandiendo el producto.

CAPÍTULO 6

Conclusiones

En este trabajo de final de máster se han sentado las bases a seguir para el desarrollo de un sistema multiagente que contenga agentes representativos de modelos de perfiles de usuario en redes sociales, con la intención de simular su comportamiento. La herramienta desarrollada consiste en la generación de un plugin para la plataforma Elgg, que permitirá la comunicación con el segundo producto resultado de este trabajo, que se trata de un sistema multiagente desarrollado sobre la plataforma SPADE. Para el desarrollo de este trabajo ha sido necesario generar mecanismos de comunicación nuevos que permitan a ambas plataformas entenderse, así como el modelado e implementación de una serie de agentes basándonos en diferentes criterios entre los que se incluyen: la edad, el género, su personalidad, su motivación e interés en realizar una acción, etc.

Respecto a los objetivos planteados en este trabajo, se puede afirmar que se han completado de manera satisfactoria siguiendo el plan de trabajo propuesto. Cabe remarcar que, debido a la gran carga de trabajo que supone un proyecto de esta envergadura, ha resultado imposible realizar un proceso de implementación que represente la totalidad de acciones que podrían realizar los agentes a causa de las restricciones temporales propias de este tipo de proyecto.

Inicialmente se realizó un proceso exhaustivo de investigación, debido a las características innovadoras de este trabajo. Por ello fue necesario consultar diferentes artículos científicos que englobaban áreas de diversa índole: psicología, tecnología, sociología, etc. Esta investigación permitió determinar el camino a seguir en los sucesivos capítulos, proporcionando en primer lugar una idea global acerca de las funcionalidades disponibles en herramientas similares, las mejores plataformas sobre las que realizar el trabajo y las propuestas de vanguardia en cuanto a perfiles de usuario, patrones de uso de redes sociales y cómo afecta su experiencia en cuestiones de privacidad a los mismos.

A continuación se ha diseñado un proceso de análisis exhaustivo en el que se han especificado, tanto formal como informalmente, las características principales del sistema, tanto a nivel funcional como a nivel conceptual, identificando los diferentes tipos de usuarios y qué acciones podrán realizar, qué componentes y medios vamos a necesitar a la hora de desplegar las soluciones desarrolladas, los diferentes problemas que podríamos encontrarnos en este proceso, las implicaciones éticas que puede tener una plataforma de este estilo, el plan de trabajo a seguir y los recursos necesarios, etc.

En la fase de implementación se han detallado los diferentes pasos seguidos para lograr sentar las bases sobre la que continuar desarrollando la plataforma. Como ya se ha comentado y se ha podido inferir durante la lectura del documento, este proyecto es muy ambicioso y su realización completa es inviable teniendo en cuenta las limitaciones temporales y de recursos propias de un trabajo de final de máster. Por este motivo se decidió desarrollar únicamente las partes más representativas y que permitirán conti-

nuar con el posterior desarrollo de este trabajo: los mecanismos de comunicación entre las plataformas junto a una acción que lo ejemplifique y la comunicación dentro del sistema multiagente que permita simular comportamientos que, en el mundo real, ocurrirían fuera de la red social. Con estos ejemplos consideramos que se cubren las partes más críticas del trabajo ya que, en el resto de casos, las acciones a realizar se basarán siempre en modificaciones sobre algunos de estos casos base, por lo que no aporta un valor real para el desarrollo del mismo.

Finalmente se ha llevado a cabo un proceso de evaluación de las soluciones desarrolladas, midiendo tanto su impacto en el sistema como su eficacia a la hora de alcanzar los resultados esperados, empleando diferentes métricas y gráficas para representar los resultados alcanzados.

Este proyecto ha supuesto un desafío en diferentes niveles, tanto a nivel de aprendizaje como a nivel tecnológico. Si bien se han utilizado conocimientos adquiridos durante los estudios (como se mencionará en la siguiente sección) ha sido necesario aprender multitud de nuevas tecnologías, metodologías y conceptos con los que no había familiaridad o no existía una base de conocimiento previa.

El primero de estos obstáculos surgió a la hora de analizar las herramientas similares. Debido a las características de este trabajo, no resultó posible encontrar herramientas que tengan una funcionalidad idéntica, por tanto fue necesario recurrir a herramientas que, si bien utilizan algunas de las tecnologías empleadas en este trabajo, no se ceñían a las mismas funcionalidades.

El segundo obstáculo surgió en el momento de buscar modelos sobre los que basarnos para la realización de los agentes. Si bien existen multitud de estudios que analizan al ser humano a nivel psicológico o sociológico, que analizan sus patrones de comportamiento en internet y en las redes sociales y cómo influyen su percepción de la privacidad en dichos patrones, no hay ningún trabajo previo que trate de relacionar todos estos conceptos en un único conjunto, por lo que ha sido necesario un profundo trabajo de investigación que permita diseñar dichos modelos.

A nivel de análisis, si bien no se han encontrado excesivas dificultades debido a la experiencia previa tanto académica como profesional, cabe remarcar la toma de decisiones seguida a la hora de desplegar la infraestructura del sistema, haciendo especial énfasis en la integración de los diferentes componentes, siendo la propuesta generada una aportación trascendente sobre la que continuar trabajando.

A nivel tecnológico se han encontrado dos grandes obstáculos. El primero de ellos surgió a la hora de desarrollar el sistema multiagente, para el que se ha necesitado aprender tanto los diferentes conceptos propios de un sistema de este tipo (no vistos previamente ni en el grado ni en el máster) como el funcionamiento de la herramienta SPADE y el lenguaje de programación Python en general. En cuanto al servidor web, a pesar de tener experiencia previa utilizando las diferentes tecnologías, ha sido necesario familiarizarse con la plataforma Elgg, sobre la que no se tenía conocimiento previo.

El último gran obstáculo, ya mencionado, ha sido el tiempo, que ha obligado a modificar los deseos iniciales de implementación para limitarlo al desarrollo de las acciones clave que permitan representar las ideas propuestas en el trabajo.

6.1 Relación del trabajo desarrollado con los estudios cursados

Para la realización de este trabajo se han utilizado conocimientos provenientes de los estudios cursados tanto en el grado como en el máster. A continuación se detallan las asignaturas de máster que han resultado más útiles:

- Para la planificación del proyecto se han empleado los conocimientos adquiridos en las asignaturas de "Gestión y gobierno de las TI", "Planificación y dirección de proyecto de TI"
- Para el diseño de la infraestructura se han empleado conocimientos extraídos de las asignaturas "Redes y seguridad", "Configuración y optimización de sistemas de cómputo" y "Servicios y aplicaciones distribuidas".
- Para la implementación se han utilizado conocimientos extraídos de las asignaturas "Computación de altas prestaciones" y "Sistemas inteligentes", aunque en esta parte del proyecto es en la que más se ha necesitado aprendizaje de forma autodidacta.
- Para la validación del sistema se han utilizado los conocimientos adquiridos en las asignaturas de "Auditoría, calidad y gestión de sistemas de información" y de "Ciencia de datos" para la representación de la información.

En cuanto a la relación de competencias transversales:

- CT-01. Comprensión e integración: ha sido necesario adquirir conocimiento de diversas fuentes y realizar una integración del mismo para generar una serie de propuestas propias.
- CT-02. Aplicación y pensamiento práctico: ha sido necesario realizar un exhaustivo proceso de análisis previa implementación.
- CT-03. Análisis y resolución de problemas: ha sido necesario resolver diferentes desafíos a nivel tecnológico, explicados con detalle a lo largo del documento.
- CT-04. Innovación, creatividad y emprendimiento: el proyecto tiene un carácter innovador incuestionable. Aunque el mismo no tenga un carácter comercial se ha decidido hacer un análisis de mercado para tratar de comprobar su situación y viabilidad.
- CT-05. Diseño y proyecto: se han utilizado técnicas de gestión de proyectos para la realización del análisis del trabajo.
- CT-06. Trabajo en equipo y liderazgo: El proyecto ha sido realizado en solitario, por lo que esta competencia no ha sido necesaria.
- CT-07. Responsabilidad ética, medioambiental y profesional: se ha realizado un análisis de las implicaciones éticas del proyecto y se ha tratado en todo momento de ceñirse ese planteamiento.
- CT-08. Comunicación efectiva: a nivel escrito. Se ha tratado de respetar los diferentes convenios y formatos sugeridos durante el proceso de redacción de este trabajo.
- CT-09. Pensamiento crítico: se ha realizado un profundo proceso de análisis, tanto durante el desarrollo del estado del arte como durante el análisis del estado actual del proyecto, tratando de argumentar en todo momento la toma de decisiones.
- CT-10. Conocimiento de problemas contemporáneos: este trabajo se encuentra englobado dentro de un proyecto que trata de resolver un problema vigente como es la falta de formación y conocimiento de los usuarios en aspectos relativos a la privacidad de sus datos y el alcance de los mismos.
- CT-11. Aprendizaje permanente: para el desarrollo del proyecto ha sido necesario aprender multitud de tecnologías y conceptos sobre los que no se tenía un conocimiento previo.

- CT-12. Planificación y gestión del tiempo: si bien se ha tratado de definir un plan de trabajo que se ciñese al número de horas exigidas por un trabajo de final de máster, el resultado final ha requerido un número de horas sustancialmente superior.
- CT-13. Instrumental específica: el proyecto hace uso de diferentes tecnologías específicas y ha requerido un gran esfuerzo en la integración de las mismas.

6.2 Trabajos futuros

Como conclusión de este trabajo van a analizarse las posibles mejoras y/o ampliaciones que puedan darse en base a este proyecto como trabajos futuros.

La primera mejora, y más obvia, es la inclusión de nuevas actividades. Si bien este proceso puede resultar aparentemente sencillo basándose en las actividades ya desarrolladas, cada nueva actividad requiere un profundo esfuerzo a la hora de realizar el análisis y la implementación de la misma ya que, a pesar de compartir un flujo de trabajo común, el proceso a realizar por cada una es muy distinto.

El segundo punto de mejora es la transferencia de la herramienta a una infraestructura de tipo cloud, que permita dotar al sistema de alta disponibilidad real. Aunque en este trabajo se proporciona un diseño utilizando tecnología de contenedores, sobre la que poder basarse a la hora de realizar esta transferencia, es evidente que el cambio a un sistema cloud implicará al responsable una serie de desafíos y toma de decisiones de complejidad notable, pudiendo incluso requerir modificaciones en el sistema de comunicación entre los subsistemas.

Como ya se ha comentado, este proyecto forma parte de un proyecto de investigación de mayor calado y, por ello, se va a utilizar como base sobre la que continuar desarrollando y ya se han planteado multitud de ampliaciones posibles. La primera ampliación consistirá en la generación de un marco normativo basado en emergencia, desarrollado como parte de mi futura tesis doctoral. Esta mejora se centrará en una ampliación profunda del sistema multiagente desarrollado en este trabajo y hará uso de los diferentes modelos propuestos con la intención de hacer a cada tipo de agente aún más único y autónomo.

Otra ampliación ya aprobada consistirá en la integración de las diferentes soluciones - desarrolladas por otros miembros del grupo de investigación - en este sistema multiagente, obteniendo así un ecosistema de desarrollo único, mucho más lógico y razonable que el actual y que permita comunicarse con los usuarios de manera más orgánica.

Una ampliación muy interesante consistiría en la inclusión del estado emocional en el comportamiento de los agentes, analizando cómo influye el mismo en el comportamiento de un usuario y trasladando ese conocimiento a las acciones de los agentes. Esta posible ampliación ofrecería una oportunidad de innovación relevante y sin duda resultaría un desafío tanto a nivel tecnológico como investigador.

De cara a la integración entre PESEDIA y el sistema multiagente, cabe recordar que no se han realizado pruebas del impacto que generarían multitud de agentes en la red social, ya que se ha optado por no implementar ninguna acción que lo represente debido a las restricciones ya mencionadas. Es vital que se valide esta casuística ya que muy posiblemente será necesario ajustar el comportamiento de los agentes para limitar sus interacciones con la red social.

Finalmente se propone el diseño de nuevos agentes que, aprovechando el sistema de comunicación generado en este trabajo, traten de realizar distintas labores de moni-

torización en el sistema con el objetivo de proporcionar a los usuarios una ayuda más personalizada que la ofrecida en la actualidad.

Bibliografía

- [1] M.-d.-C. Alarcón-del Amo, C. Lorenzo-Romero, and M.-Á. Gómez-Borja. Classifying and profiling social networking site users: A latent segmentation approach. *Cyberpsychology, behavior, and social networking*, 14(9):547–553, 2011.
- [2] J. Alemany Bordera. Pesedia. red social para concienciar en privacidad. 2016.
- [3] Y. Amichai-Hamburger and E. Ben-Artzi. Loneliness and internet use. *Computers in human behavior*, 19(1):71–80, 2003.
- [4] Y. Amichai-Hamburger and G. Vinitzky. Social network use and personality. *Computers in human behavior*, 26(6):1289–1295, 2010.
- [5] T. Amiel and S. L. Sargent. Individual differences in internet usage motives. *Computers in Human Behavior*, 20(6):711–726, 2004.
- [6] E. Argente, E. Vivancos, J. Alemany, and A. García-Fornes. Educando en privacidad en el uso de las redes sociales. *Education in the Knowledge Society*, 18(2):107–126, 2017.
- [7] J. B. Asendorpf, P. Borkenau, F. Ostendorf, and M. A. Van Aken. Carving personality description at its joints: Confirmation of three replicable personality prototypes for both children and adults. *European Journal of Personality*, 15(3):169–198, 2001.
- [8] S. B. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), 2006.
- [9] F. Bellifemine, A. Poggi, and G. Rimassa. Jade—a fipa-compliant agent framework. In *Proceedings of PAAM*, volume 99, page 33. London, 1999.
- [10] J. O. Berndt, S. C. Rodermund, F. Lorig, and I. J. Timm. Modeling user behavior in social media with complex agents. In *Third International Conference on Human and Social Analytics*, pages 18–24, 2017.
- [11] L. Braubach, A. Pokahr, and W. Lamersdorf. Jadex: A short overview. In *Main Conference Net. ObjectDays*, volume 2004, pages 195–207, 2004.
- [12] C. Casado Riera, U. Oberst, and X. Carbonell. Facebook: personalidad y privacidad en los perfiles anuario de psicología, vol. 45, núm. 1, abril, 2015, pp. 39-54 universitat de barcelona barcelona, españa. *Anuario de Psicología*, 45(1):39–54, 2015.
- [13] A. Caspi and P. A. Silva. Temperamental qualities at age three predict personality traits in young adulthood: Longitudinal evidence from a birth cohort. *Child development*, 66(2):486–498, 1995.
- [14] E. Constantinides, M. Carmen Alarcón del Amo, and C. L. Romero. Profiles of social networking sites users in the netherlands. 2010.

- [15] T. Correa, A. W. Hinsley, and H. G. De Zuniga. Who interacts on the web?: The intersection of users' personality and social media use. *Computers in human behavior*, 26(2):247–253, 2010.
- [16] P. T. Costa Jr, J. H. Herbst, R. R. McCrae, J. Samuels, and D. J. Ozer. The replicability and utility of three personality types. *European Journal of Personality*, 16(S1):S73–S87, 2002.
- [17] S. Das and A. Kramer. Self-censorship on facebook. In *Seventh international AAAI conference on weblogs and social media*, 2013.
- [18] B. O. del Estado. Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. 2018.
- [19] H. Donâncio, A. Casals, and A. A. Brandao. Exposing agents as web services: a case study using jade and spade. 2019.
- [20] M. B. Donnellan and R. W. Robins. Resilient, overcontrolled, and undercontrolled personality types: Issues and controversies. *Social and Personality Psychology Compass*, 4(11):1070–1083, 2010.
- [21] A. Ehrenberg, S. Juckes, K. M. White, and S. P. Walsh. Personality and self-esteem as predictors of young people's technology use. *Cyberpsychology & behavior*, 11(6):739–741, 2008.
- [22] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini. The rise of social bots. *Commun. ACM*, 59(7):96–104, jun 2016.
- [23] M. Gerlach, B. Farb, W. Reville, and L. A. N. Amaral. A robust data-driven approach identifies four personality types across four large data sets. *Nature Human Behaviour*, 2(10):735, 2018.
- [24] L. R. Goldberg. An alternative "description of personality": the big-five factor structure. *Journal of personality and social psychology*, 59(6):1216, 1990.
- [25] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.
- [26] O. Isbulan. Opinions of university graduates about social networks according to their personal characteristics. *Turkish Online Journal of Educational Technology-TOJET*, 10(2):184–189, 2011.
- [27] P. Karr-Wisniewski, D. Wilson, and H. Richter-Lipford. A new social order: Mechanisms for social network site boundary regulation. In *Americas Conference on Information Systems, AMCIS*, 2011.
- [28] K. Kravari and N. Bassiliades. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1):11, 2015.
- [29] A. Lenhart and M. Madden. Social networking websites and teens: An overview. 2007.
- [30] K. T. Li-Barber. Self-disclosure and student satisfaction with facebook. *Computers in Human behavior*, 28(2):624–630, 2012.
- [31] M. Madden. Privacy management on social media sites. *Pew Internet Report*, pages 1–20, 2012.

- [32] M. Madden, A. Lenhart, S. Cortesi, U. Gasser, M. Duggan, A. Smith, and M. Beaton. Teens, social media, and privacy. *Pew Research Center*, 21:2–86, 2013.
- [33] J. Maria Balmaceda, S. Schiaffino, and D. Godoy. How do personality traits affect communication among users in online social networks? *Online Information Review*, 38(1):136–153, 2014.
- [34] K. Moore and J. C. McElroy. The influence of personality on facebook usage, wall postings, and regret. *Computers in Human Behavior*, 28(1):267–274, 2012.
- [35] P. A. Norberg, D. R. Horne, and D. A. Horne. The privacy paradox: Personal information disclosure intentions versus behaviors. *Journal of consumer affairs*, 41(1):100–126, 2007.
- [36] M. Obitko. Fipa compliant agents, 2007. [Online; accessed 05-June-2019].
- [37] P. D. O’Brien and R. C. Nicol. Fipa—towards a standard for software agents. *BT Technology Journal*, 16(3):51–59, 1998.
- [38] E. Y. Ong, R. P. Ang, J. C. Ho, J. C. Lim, D. H. Goh, C. S. Lee, and A. Y. Chua. Narcissism, extraversion and adolescents’ self-presentation on facebook. *Personality and individual differences*, 50(2):180–185, 2011.
- [39] U. Pfeil, R. Arjan, and P. Zaphiris. Age differences in online social networking—a study of user profiles and the social capital divide among teenagers and older users in myspace. *Computers in Human Behavior*, 25(3):643–654, 2009.
- [40] A. Pokahr, L. Braubach, and K. Jander. The jadex project: Programming model. In *Multiagent Systems and Applications*, pages 21–53. Springer, 2013.
- [41] A. Pokahr, L. Braubach, and W. Lamersdorf. Jadex: A bdi reasoning engine. In *Multi-agent programming*, pages 149–174. Springer, 2005.
- [42] T. Rendón Sallard and M. Sánchez-Marrè. A review on multi-agent platforms and environmental decision support systems simulation tools. 2006.
- [43] R. W. Robins, O. P. John, A. Caspi, T. E. Moffitt, and M. Stouthamer-Loeber. Resilient, overcontrolled, and undercontrolled boys: three replicable personality types. *Journal of Personality and Social psychology*, 70(1):157, 1996.
- [44] C. Ross, E. S. Orr, M. Sasic, J. M. Arseneault, M. G. Simmering, and R. R. Orr. Personality and motivations associated with facebook use. *Computers in human behavior*, 25(2):578–586, 2009.
- [45] T. Ryan and S. Xenos. Who uses facebook? an investigation into the relationship between the big five, shyness, narcissism, loneliness, and facebook usage. *Computers in human behavior*, 27(5):1658–1664, 2011.
- [46] E.-M. Schomakers, C. Lidynia, L. Vervier, and M. Ziefle. Of guardians, cynics, and pragmatists—a typology of privacy concerns and behavior. In *IoTBDs*, pages 153–163, 2018.
- [47] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.

-
- [48] M. Thelwall, D. Wilkinson, and S. Uppal. Data mining emotion in social network communication: Gender differences in myspace. *Journal of the American Society for Information Science and Technology*, 61(1):190–199, 2010.
- [49] K. Wilson, S. Fornasier, and K. M. White. Psychological predictors of young adults' use of social networking sites. *Cyberpsychology, behavior, and social networking*, 13(2):173–177, 2010.
- [50] M. Winikoff. JackTM intelligent agents: an industrial strength platform. In *Multi-Agent Programming*, pages 175–193. Springer, 2005.
- [51] P. J. Wisniewski, B. P. Knijnenburg, and H. R. Lipford. Making privacy personal: Profiling social network users to inform privacy education and nudging. *International Journal of Human-Computer Studies*, 98:95–108, 2017.
- [52] K. Yoshimura. Fipa jack: A plugin for jack intelligent agents tm. *Manual*, RMIT University, 2003.
- [53] M. Zuckerman, D. M. Kuhlman, J. Joireman, P. Teta, and M. Kraft. A comparison of three structural models for personality: the big three, the big five, and the alternative five. *Journal of personality and social psychology*, 65(4):757, 1993.