



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

# Diseño e implementación de una plataforma de creación de mercados virtuales basados en sistemas multiagente

TRABAJO DE FINAL DE GRADO

Grado en Ingeniería Informática

*Autor:* Mejías Rodríguez, José Manuel

*Tutor:* Garrido Tejero, Antonio

2 de septiembre de 2014

## Resumen

Con el auge en los últimos años de los mercados electrónicos y la formación de diversas empresas de comercio electrónico surge la necesidad de automatizar procesos de negocio mediante herramientas informatizadas. En la actualidad, múltiples soluciones informáticas permiten la creación de plataformas web orientadas a la compraventa de productos electrónicos, tales como eBay, Amazon y uShip, entre otros. Sin embargo no se ha resuelto la necesidad de automatizar el alcance de acuerdos para la compraventa de productos y servicios.

Los sistemas multiagente, formados por diversas entidades software que muestran propiedades de persistencia, autonomía, reactividad y habilidad social (también denominados agentes inteligentes), facilitan el alcance de acuerdos en procesos colaborativos sin la intervención humana para alcanzar los objetivos individuales de los miembros del sistema multiagente. En el contexto de un mercado electrónico, dichos sistemas pueden facilitar el alcance de acuerdos para la compraventa de productos y servicios de forma autónoma, controlada por los objetivos del usuario humano que los representa y resultan especialmente útiles en entornos en los que es necesario tratar un gran volumen de posibles acuerdos.

Con el objetivo de demostrar las capacidades de dichos sistemas para el problema propuesto se pretende el diseño e implementación de un prototipo que permita emular el funcionamiento de un mercado físico con una naturaleza genérica tanto en los productos y servicios que se puedan comercializar como del protocolo empleado para alcanzar el acuerdo (protocolos tales como la subasta inglesa, la subasta holandesa, la subasta japonesa, la compra directa o la negociación cara a cara) mediante el uso de las tecnologías del acuerdo anteriormente descritas. Para ello se emplearán metodologías de diseño de dichos sistemas multiagente como la metodología ROMAS y plataformas para la gestión de servicios orientados a dichos sistemas como la plataforma MAGENTIX2.

*Palabras clave:* sistemas multiagente, organizaciones virtuales, mercados, comercio, inteligencia artificial, negociación automática

# Índice general

<b>1. Introducción</b>	<b>10</b>
1.1. Descripción del problema . . . . .	10
1.2. Contexto . . . . .	11
1.2.1. Contexto socio-económico . . . . .	11
1.2.2. Grupo de Tecnología Informática - Inteligencia Artificial . . . . .	12
1.3. Motivación y alcance del proyecto . . . . .	13
1.4. Objetivos . . . . .	15
<b>2. Estado del arte</b>	<b>17</b>
2.1. Mercados . . . . .	17
2.1.1. Mercados laborales . . . . .	18
2.1.2. Mercados virtuales . . . . .	18
2.1.3. Plataformas de mercado . . . . .	19
2.2. Agentes software y sistemas multiagente . . . . .	20
2.2.1. Definición de agente . . . . .	20
2.2.2. Sistemas multiagente . . . . .	20
2.3. Tecnologías del acuerdo . . . . .	21
2.3.1. Negociación automática . . . . .	22
<b>3. Análisis y diseño</b>	<b>23</b>
3.1. Modelo de negocio . . . . .	23
3.1.1. Propuesta de valor . . . . .	25
3.1.2. Segmentos de cliente . . . . .	25
3.1.3. Canales de distribución . . . . .	27
3.1.4. Relaciones con clientes . . . . .	27

---

3.1.5. Fuentes de ingresos . . . . .	28
3.1.6. Recursos clave . . . . .	28
3.1.7. Actividades clave . . . . .	28
3.1.8. Socios clave . . . . .	29
3.1.9. Estructura de costes . . . . .	29
3.2. Análisis y diseño del sistema mediante la metodología ROMAS . . . . .	29
3.2.1. Metodología de diseño (ROMAS) . . . . .	29
3.2.2. Análisis y diseño del sistema . . . . .	32
<b>4. Implementación</b>	<b>71</b>
4.1. Introducción . . . . .	71
4.2. Java . . . . .	73
4.3. MAGENTIX2 . . . . .	73
4.4. Otras herramientas . . . . .	75
4.5. Del diseño a la implementación . . . . .	77
4.5.1. Diagrama de clases . . . . .	77
4.5.2. Diseño de la interfaz gráfica de usuario . . . . .	78
4.6. Prototipo . . . . .	91
4.6.1. Diseño del prototipo . . . . .	97
4.6.2. Implementación del prototipo . . . . .	99
<b>5. Conclusiones y trabajo futuro</b>	<b>111</b>
5.1. Conclusiones . . . . .	111
5.2. Trabajo futuro . . . . .	112
<b>Apéndices</b>	<b>116</b>
<b>A. Diagramas ilustrativos del desglose de objetivos en tareas</b>	<b>117</b>

# Índice de figuras

3.1. <i>Business model canvas</i> del proyecto . . . . .	26
3.2. Diagrama ilustrativo de las fases de la metodología . . . . .	31
3.3. Diagrama de objetivos del sistema . . . . .	36
3.4. Diagrama de secuencia desde un punto de vista general . . . . .	37
3.5. Diagrama de secuencia orientado a la gestión de usuarios de la plataforma	38
3.6. Diagrama de secuencia orientada a la gestión del mercado . . . . .	40
3.7. Diagrama de secuencia orientada a la interacción con el mercado . . .	41
3.8. Diagrama que resume la funcionalidad del diseñador de la plataforma de mercado . . . . .	44
3.9. Diagrama que resume la funcionalidad del usuario no registrado en la plataforma del mercado . . . . .	44
3.10. Diagrama que resume la funcionalidad del usuario registrado en la plataforma del mercado . . . . .	45
3.11. Diagrama que resume la funcionalidad del gestor de la plataforma de mercado . . . . .	46
3.12. Diagrama que resume la funcionalidad del tasador de la plataforma de mercado . . . . .	46
3.13. Diagrama que resume la funcionalidad del moderador de la plataforma de mercado . . . . .	47
3.14. Diagrama que resume la funcionalidad del controlador de mesas de negociación de la plataforma de mercado . . . . .	47
3.15. Diagrama que resume la funcionalidad del juez de mesa de negociación de la plataforma de mercado . . . . .	48
3.16. Diagrama que resume la funcionalidad del usuario del mercado . . . .	49

---

3.17. Estructura social y organizacional . . . . .	50
3.18. Diagrama que indica la interacción entre los distintos roles para la realización de tareas . . . . .	51
3.19. Diagrama de interacción correspondiente al registro de un usuario . .	54
3.20. Diagrama de interacción correspondiente a la identificación de usuarios	55
3.21. Diagrama de interacción correspondiente al proceso después de la iden- tificación de un usuarioki . . . . .	56
3.22. Diagrama de interacción correspondiente a la solicitud de una tasación	57
3.23. Diagrama de interacción correspondiente a la petición del listado de peticiones de tasado . . . . .	58
3.24. Diagrama de interacción correspondiente a la solicitud de los detalles de una petición de tasación . . . . .	58
3.25. Diagrama de interacción correspondiente a la resolución de una petición de tasación . . . . .	58
3.26. Diagrama de interacción correspondiente a la petición de la apertura de una mesa de negociación . . . . .	59
3.27. Diagrama de interacción correspondiente a la solicitud de los detalles de una petición de apertura de una mesa de negociación . . . . .	60
3.28. Diagrama de interacción correspondiente a la notificación de la resolu- ción de una petición de apertura de una mesa de negociación . . . . .	60
3.29. Diagrama de interacción correspondiente a la notificación de cambio del protocolo de resolución de peticiones de apertura de mesas de negociación	61
3.30. Diagrama de interacción correspondiente a la petición de la lista de mesas de negociación abiertas . . . . .	62
3.31. Diagrama de interacción correspondiente a la solicitud de los detalles de una mesa de negociación específica . . . . .	63
3.32. Diagrama de interacción correspondiente a la negociación genérica de acuerdos . . . . .	64
3.33. Diagrama de interacción correspondiente al protocolo de negociación de subasta japonesa . . . . .	65
3.34. Diagrama de interacción correspondiente a la notificación del cambio en el estado del acuerdo . . . . .	66

3.35. Diagrama de interacción correspondiente a la petición de la lista de conflictos . . . . .	67
3.36. Diagrama de interacción correspondiente a la solicitud de los detalles de un conflicto específico . . . . .	67
3.37. Diagrama de interacción correspondiente a la notificación de la resolución de un conflicto . . . . .	68
3.38. Diagrama de interacción correspondiente a la parada de agentes debido a la pausa/parada de la actividad de la plataforma de mercado . . . .	69
3.39. Diagrama de interacción correspondiente al envío de una señal de desactivación de agente . . . . .	69
4.1. Arquitectura del mercado genérico virtual implementado . . . . .	72
4.2. Logotipo y esquema ilustrativo de las funcionalidades de la plataforma MAGENTIX2 . . . . .	75
4.3. Diagrama de clases correspondiente a la plataforma de mercado . . .	79
4.4. Prototipo de la interfaz correspondiente a la pantalla de bienvenida al usuario . . . . .	80
4.5. Prototipo de la interfaz correspondiente al registro del usuario . . . .	80
4.6. Prototipo de la interfaz correspondiente al panel de administración del gestor del mercado . . . . .	81
4.7. Prototipo de la interfaz correspondiente a la información del mercado	82
4.8. Prototipo de la interfaz correspondiente a la visualización de estadísticas del mercado . . . . .	82
4.9. Prototipo de la interfaz correspondiente a la gestión del ciclo de vida del mercado . . . . .	83
4.10. Prototipo de la interfaz correspondiente a al panel de administración de usuarios del mercado . . . . .	84
4.11. Prototipo de la interfaz correspondiente al registro manual de usuarios por parte del gestor de mercado . . . . .	85
4.12. Prototipo de la interfaz correspondiente a la gestión de peticiones de registro de usuarios . . . . .	86
4.13. Prototipo de la interfaz correspondiente a la visualización de peticiones de suspensión de usuarios . . . . .	86

---

4.14. Prototipo de la interfaz correspondiente al panel para modificar los roles disponibles para cada usuario del mercado . . . . .	87
4.15. Prototipo de la interfaz correspondiente a la visualización y resolución de las peticiones de tasado de productos . . . . .	88
4.16. Prototipo de la interfaz correspondiente a la visualización de peticiones de moderación de conflictos . . . . .	88
4.17. Prototipo de la interfaz correspondiente a la resolución de un conflicto por parte de un moderador . . . . .	89
4.18. Prototipo de la interfaz correspondiente a la gestión de las peticiones de apertura de las mesas de negociación . . . . .	90
4.19. Prototipo de la interfaz correspondiente al panel de uso del mercado a un usuario . . . . .	92
4.20. Prototipo de la interfaz correspondiente a la visualización de las mesas de negociación del mercado . . . . .	92
4.21. Prototipo de la interfaz correspondiente a la petición de apertura de una mesa de negociación . . . . .	93
4.22. Prototipo de la interfaz correspondiente a la visualización de una mesa de negociación y las acciones a realizar . . . . .	93
4.23. Prototipo de la interfaz correspondiente a la definición de un producto	94
4.24. Prototipo de la interfaz correspondiente a la visualización de las peticiones de tasación realizadas . . . . .	94
4.25. Prototipo de la interfaz correspondiente a la gestión de los acuerdos realizados y su estado . . . . .	95
4.26. Prototipo de la interfaz correspondiente a la gestión del agente representante del usuario . . . . .	96
4.27. Prototipo de la interfaz correspondiente a la modificación de los datos del usuario . . . . .	97
4.28. Listado de clases generadas mediante Hibernate . . . . .	100
4.29. Código generado de forma automática para la entidad Usuario . . . . .	101
4.30. Extracto de código correspondiente al <i>Market Manager</i> . . . . .	103
4.31. Extracto de código correspondiente al registro de un usuario en la base de datos . . . . .	104



4.32. Extracto de código correspondiente al <i>Table Controller</i> . . . . .	105
4.33. Listado de métodos implementados en la clase <code>MarketUser.java</code> . . . . .	106
4.34. Extracto de código que representa parte la interacción con el usuario	107
4.35. Extracto de código que representa la interacción con el administrador de mercado para el registro de un usuario . . . . .	108
4.36. Interacción entre el usuario y el sistema multiagente . . . . .	109
4.37. Interacción entre el usuario y el sistema multiagente (continuación) .	110
A.1. Diagrama de tareas referente a la definición del mercado . . . . .	118
A.2. Diagrama de tareas referente a la administración de usuarios . . . . .	119
A.3. Diagrama de tareas referente a la administración del mercado . . . . .	119
A.4. Diagrama de tareas referente a la interacción con el mercado . . . . .	120

# Índice de cuadros

3.1. Descripción general del sistema propuesto . . . . .	34
3.2. Ejemplo de normas generadas para la plataforma . . . . .	53

# Capítulo 1

## Introducción

En este capítulo se introducirá el problema que se desea resolver mediante una breve descripción, la motivación surgida a partir del contexto actual para la realización del trabajo y los objetivos propuestos a cumplir durante el desarrollo del proyecto.

### 1.1. Descripción del problema

En la sociedad en la que nos situamos nos encontramos con situaciones en la que necesitamos adquirir ciertos productos y/o servicios para poder subsistir. Ya seamos una empresa o una entidad particular existen productos que necesitamos para persistir ya sean de alimentación, ocio o necesarios para progresar en procesos de negocio. Existe la tendencia de reducir costes mediante la compra de estos productos en mercados ya sean físicos o electrónicos. Es por ello por lo que existen comparadores por Internet que permiten comparar distintas ofertas y reducir el esfuerzo de recopilar información. En el contexto de un mercado, este proceso puede involucrar tanto al comprador como al vendedor.

Si generalizamos esta descripción observamos como podemos adaptar esta mentalidad para otras situaciones como la búsqueda de empleo como se va empleando en distintas plataformas empleadas en la actualidad como *LinkedIn* (<http://www.linkedin.com>) u *Infojobs* (<http://www.infojobs.net>). El control de la mediación en conflictos o la petición de servicios de tasado de productos o asesoramiento relacionado con los mismos.

Por otra parte, la automatización ha surgido como vía para reducir el esfuerzo

requerido por personas y empresas para comprar determinados tipos de productos o contratar determinados tipos de servicios. Sin embargo, se observa que no existen herramientas que permitan llegar a realizar estas compras (y ventas) de forma automática sin la gestión de un usuario durante todo el proceso. Esta situación puede llegar a agravarse en casos en la que exista una gran cantidad de compradores y vendedores, los productos sean muy distintos o las capacidades del comprador y vendedor no permitan llegar a realizar estas acciones de forma coherente tal como puede ocurrir con personas discapacitadas o empresas emergentes en mercados en los que no dispone de visibilidad.

Es por ello por lo que se propone resolver estos problemas mediante la creación de una herramienta que permita facilitar la mediación entre compradores y vendedores. Esta mediación y negociación de las partes puede ser realizada de forma automática o de forma semiautomática en base a los objetivos del usuario humano al que representa. Dadas las limitaciones de los sistemas informáticos en la actualidad y la penetración de Internet en la sociedad se propone que esta plataforma se controle mediante medios electrónicos empleando esta vía de comunicación. Por último, se propone además ofrecer servicios adicionales a estas entidades para facilitar la gestión de estos procesos de mediación, tales como la tasación de productos.

## 1.2. Contexto

En esta sección se introduce al lector al contexto socio-económico con la intención de justificar la realización del presente trabajo y las tecnologías que más han influido en las decisiones tomadas para su realización.

### 1.2.1. Contexto socio-económico

Si observamos la evolución del contexto socio-económico en los últimos años en el campo de la Administración y dirección de Empresas se observa la aparición de políticas empresariales potenciadas por la aparición durante el año 2008 de una crisis económica en las cuales las empresas intentan reducir costes sin afectar a la calidad de los productos ofrecidos mediante la optimización de sus procesos de negocio.

Entre los departamentos en los que se plantea dicha optimización se encuentra

por un lado la compraventa de bienes y la contratación de servicios, en las que las empresas dedican gran parte de sus recursos en diversos departamentos con el objetivo de adquirir los productos necesarios en los procesos de negocio, contratar los servicios necesarios en dichos procesos, vender los productos generados por la empresa con el objetivo de obtener beneficios y ofrecer los servicios que puede realizar la empresa a terceros.

Otro de los departamentos en el que las empresas dedican recursos para obtener más mano de obra dentro de la empresa y un mejor entorno de trabajo en el que ofrecer sus productos y servicios son los recursos humanos. Con la aparición de la crisis económica se han producido una gran cantidad de recortes en las plantillas de las empresas. Es por ello por lo que una correcta gestión de los recursos humanos de la empresa permite reducir el número de despidos en situaciones críticas mediante el desarrollo de planes de contingencia más justos y adecuados además de facilitar la tarea de la contratación de nuevo personal.

Dada la importancia crítica de obtener los mejores acuerdos en estas tareas para mejorar los beneficios y garantizar la supervivencia de la empresa se dedica una gran cantidad de recursos humanos y materiales. Diversos perfiles profesionales han surgido a raíz de dichas necesidades además de herramientas software como los Planificadores de Recursos Empresariales (en inglés *Enterprise Resource Planning* o *ERP*) o plataformas para la automatización de demandas de recursos tanto humanos como materiales orientado a publicitar a empresas externas.

Para ello es necesario suplir las limitaciones en las cuales las empresas pueden asignar dichos recursos además de las limitaciones humanas en el tratamiento de estos casos, tales como la necesidad en algunos sectores orientados a la venta por Internet de disponer de un empleado gestionando acuerdos las 24 horas del día, los siete días de la semana y, en algunos casos, los 365 días del año tales como el mercado de servicios *freelance*.

### **1.2.2. Grupo de Tecnología Informática - Inteligencia Artificial**

Desde el Grupo de Tecnología Informática - Inteligencia Artificial, ubicado en el Departamento de Sistemas Informáticos y Computación en la Universidad Politécnica

de Valencia se han desarrollado líneas de investigación orientadas al tratamiento de problemas reales mediante el uso de tecnologías para su aplicación en campos como el control de procesos, el control de entornos industriales complejos, la búsqueda y recuperación de información y el comercio electrónico. Para ello se han desarrollado metodologías y entornos de desarrollo orientados al campo de los sistemas multiagente y las tecnologías del acuerdo, conceptos que se desarrollarán a lo largo de este documento.

Enmarcado en el desarrollo en dichas tecnologías el grupo ha desarrollado una plataforma que permite el desarrollo de aplicaciones orientadas a sistemas multiagente abiertos denominada MAGENTIX2 [2] en la cual se ofrecen servicios tales como trazas, agentes conversacionales, agentes argumentativos y agentes Jason. Además, para facilitar el análisis y diseño de tales sistemas, el grupo ha desarrollado metodologías tales como la metodología ROMAS ([8],[9]) que permite facilitar la modelización de dichos sistemas en un proceso secuencial generando finalmente un esquema para el desarrollo posterior de dichos sistemas incluyendo un control normativo orientado a los agentes que pertenecen a los mismos.

El trabajo realizado corresponde al resultado del desarrollo realizado dentro del grupo de investigación a través de una beca de colaboración durante el curso 2013/14.

### **1.3. Motivación y alcance del proyecto**

Observando las necesidades de las empresas en el contexto socio-económico surge una necesidad de automatizar los procesos de compraventa de bienes y recursos humanos. Se pretende que las empresas puedan alcanzar acuerdos de una manera rápida y efectiva en base a los objetivos de la empresa y superando las limitaciones humanas para el control de dichos acuerdos como puede ser la gestión de múltiples negociaciones simultáneas, el uso de protocolos de negociación complejos en la que los humanos podemos tener dificultades de entender o emplear y la velocidad en la que podemos realizar las negociaciones. Un ejemplo de este último hecho se puede observar en las negociaciones en plataformas en línea de subastas en las que los usuarios realizan las últimas pujas por un artículo en los últimos segundos antes del cierre de la subasta. El desarrollo de las tecnologías de sistemas multiagente y tecnologías del acuerdo

permiten el desarrollo de plataformas de agentes en las cuales se pueden alcanzar acuerdos de una forma organizada y que además de las características anteriormente mencionadas permiten controlar las acciones de los miembros del mismo mediante un control normativo dado por las restricciones de la organización a la que pertenecen y la naturaleza del mercado.

El resultado final del desarrollo del proyecto es el diseño de una plataforma de mercado en la cual sus organizadores pueden definir qué tipos de productos se pueden comerciar en él (sin importar su naturaleza), controlar mediante restricciones las acciones que pueden realizar los usuarios y qué protocolos pueden emplear para alcanzar dichos acuerdos. A partir de dicha definición se podrá lanzar a ejecución un mercado que permita a los usuarios interactuar con el mismo de forma análoga a un mercado físico con el objetivo de alcanzar acuerdos sobre la compraventa de dichos productos, formalizándose en un contrato vinculante entre las partes implicadas. Este alcance de acuerdos se podrá realizar de forma manual o bien mediante el uso de una entidad representante del usuario que alcance los acuerdos de forma automática en base a unos objetivos dados y una serie de criterios que permiten medir la utilidad de los acuerdos a la hora de escoger los mejores acuerdos posibles de acuerdo a sus posibilidades.

Por otra parte se pretende el control de los acuerdos alcanzados en la plataforma, ofreciendo servicios adicionales de moderación de conflictos en casos particulares. Finalmente, se ofrecerá a los usuarios un servicio de tasación de productos en base a una especificación del producto material. En cuanto a los gestores del mercado, se podrán monitorizar las acciones realizadas dentro de la plataforma de mercado, gestionando tanto los usuarios como el ciclo de vida de la plataforma, pudiendo pausar o parar la ejecución de la misma.

Por último y para facilitar la interacción entre los usuarios de la plataforma se plantea el diseño de una plataforma web que permita que los usuarios puedan gestionar las negociaciones y acuerdos alcanzados además de emplear los servicios que ofrece la plataforma de una manera similar a la que herramientas similares existentes en el mercado tales como *Ebay* (<http://www.ebay.com>) o *Amazon* (<http://www.amazon.com>).

## 1.4. Objetivos

El objetivo principal de este trabajo es el análisis, diseño e implementación de un entorno de desarrollo que permita facilitar la creación de plataformas de mercado con una naturaleza genérica que permita el alcance de acuerdos sobre un conjunto de bienes y servicios empleando tecnologías punteras en el marco de los sistemas multiagente y las tecnologías del acuerdo. Dicho entorno de desarrollo debe cubrir las necesidades de las empresas que participan en la creación de las plataformas de mercado, por lo que deben ser abiertas, dinámicas, flexibles y adaptables al dominio de aplicación. Este objetivo principal se desglosa en los siguientes subobjetivos:

- Análisis del estado del arte
  - Estudio del mercado de las plataformas de mercado y su impacto en la población y el ecosistema empresarial.
  - Estudio del estado del arte de las plataformas de mercado existentes, su funcionamiento y sus limitaciones.
  - Estudio de la metodología ROMAS y su posible uso para el análisis y diseño del entorno de desarrollo.
  - Estudio de un posible modelo de negocio con el cual orientar el producto final producido a su comercialización.
- Análisis y diseño del entorno de desarrollo mediante la metodología ROMAS.
  - A partir de los resultados de ambos estudios, desarrollar el modelado de la plataforma, incluyendo el modelado del sistema multiagente que permite el funcionamiento de las plataformas de mercado creadas mediante el entorno de desarrollo.
  - Además de las herramientas empleadas y de forma auxiliar para su posterior implementación, desarrollar modelos UML que permitan diseñar partes de la plataforma en sus etapas tempranas.
- Implementación del entorno de desarrollo de mercados mediante la plataforma de servicios para sistemas multiagente MAGENTIX2.



- Estudio de la plataforma de servicios para agentes MAGENTIX2, sus APIs destinadas a las necesidades del proyecto y su adecuación al trabajo.
  - Tras su estudio y en base a los resultados del estudio y del modelo generado en la fase anterior implementar un prototipo en base a la especificación que permita interactuar con un mercado genérico con funcionalidades básicas a modo de demostración.
- Evaluar los resultados obtenidos y planificación de proyectos futuros.
- En base a los objetivos cumplidos y los resultados obtenidos planificar las actuaciones futuras en base a los objetivos del grupo de investigación y de las necesidades de las empresas interesadas en las tecnologías realizadas en el presente trabajo.

En este punto finaliza el capítulo de introducción en el que se ha expuesto al lector una descripción del problema, el contexto en el que surgen una serie de necesidades que se tratan en el alcance del proyecto, una breve descripción de un posible modelo de negocio del producto final producido y los objetivos a realizar para la consecución del trabajo. El resto del trabajo se estructura de la siguiente manera. En el Capítulo 2 se detalla el estado del arte de las tecnologías empleadas en el trabajo y su relación con el mismo. En el Capítulo 3 se explica de forma detallada el proceso de análisis y diseño de la plataforma de mercado. En el Capítulo 4 se detalla el proceso de implementación de un prototipo que muestra las el proceso y las herramientas empleadas en la implementación de un prototipo que permita mostrar la funcionalidad de los sistemas multiagente para el tratamiento del problema.

# Capítulo 2

## Estado del arte

En este capítulo se introduce al lector al estado del arte de las tecnologías empleadas en el trabajo así como el desarrollo social y empresarial orientado al contexto y dominios de aplicación en los que se introduce la solución generada en el trabajo. Se hablará sobre los mercados como entidades para alcanzar acuerdos, los mercados virtuales como especialización de los mercados en un entorno digital, las plataformas de mercado como soporte para los mercados virtuales, los sistemas multiagente como soporte a las organizaciones virtuales y las tecnologías del acuerdo, entre las que se incluye la negociación automática.

### 2.1. Mercados

Se define como mercado al conjunto de transacciones de bienes y servicios entre individuos o asociaciones de individuos. Dichas transacciones involucran el uso de bienes (posesiones de las partes del mercado) y/o servicios (cumplir con las necesidades de un cliente) y pueden emplear el uso de monedas de cambio como mediación en las negociaciones que dan a lugar las transacciones. Las transacciones se pueden clasificar en función de la naturaleza de los compradores y vendedores en: de empresa a empresa (*business to business, B2B*), de empresa a cliente (*business to client, B2C*), de cliente a cliente (*client to client, C2C*) y de cliente a gobierno (*client to government, C2G*).

Dependiendo del tipo de bienes a negociar se clasifican en mercados de productos

si el elemento a negociar son bienes tangibles o intangibles que se pueden consumir; mercados de servicios si se trata de una serie de acciones englobadas en el proceso de negocio de una de las partes y mercados de labor (mercados laborales) si éstos involucran al trabajador final como agente económico y una o varias empresas o agrupaciones a las que presta servicio.

### **2.1.1. Mercados laborales**

Se define como mercado laboral al mercado en el que se involucra por una de las partes el ofrecimiento de la realización de servicios de una entidad final de un trabajador para el cumplimiento de uno o varios de los objetivos de la empresa a la que se adhiere de forma temporal mediante un contrato. Este contrato puede tener una duración definida y unas condiciones o cláusulas que limitarán los derechos, obligaciones y responsabilidades del trabajador dentro de la estructura organizativa de la empresa.

### **2.1.2. Mercados virtuales**

Con la aparición de Internet se ha comenzado a acuñar el término mercado electrónico a los mercados que discurren a través de la red. En la sociedad actual se ha observado un aumento progresivo en los últimos años en la venta de productos y servicios a través de la red. A diferencia de la década de los 90 y debido principalmente al auge del uso de dispositivos móviles y la mejora de la tecnología de las telecomunicaciones, el uso de mercados electrónicos para la compraventa de productos y servicios son algo ya cotidiano para la población. La penetración de estas tecnologías en el mercado ha dado lugar a la aparición de múltiples portales web que ofrecen la posibilidad de comprar y vender bienes de forma electrónica ofrecidos por usuarios particulares y/o vendedores profesionales. Ejemplos de estos portales son *Ebay*, *Amazon* y en el contexto de dispositivos móviles *Wallapop* (<http://www.wallapop.com/>).

Por otra parte, han surgido servicios en la red que permiten la mediación en la compra de productos y contratación de servicios. Algunos servicios incluso permiten visualizar de una manera rápida y completa las distintas posibilidades en la contratación y compra de dichos servicios. Ejemplos de dominios de aplicación donde se

pueden observar estos servicios son la contratación de seguros, paquetes vacacionales y la contratación de servicios de telefonía.

### 2.1.3. Plataformas de mercado

Con el objetivo de agrupar diversos mercados de una forma muy similar a los parques comerciales físicos se plantea la inclusión de las plataformas de mercado como concepto. Estas agrupaciones permiten agilizar y reducir los costes asociados a la administración de estas infraestructuras. En el entorno web múltiples aplicaciones han aplicado este concepto para crear entornos en los que comerciar productos y servicios. Ejemplos de estas plataformas son servicios en línea como *Ebay*, *Amazon*, *uShip* o *Freelance*.

En las plataformas de mercado el objetivo es formalizar acuerdos que impliquen la compraventa de productos o la contratación de servicios determinados (incluyendo los contratos laborales). Dichos acuerdos se formalizan de forma legal con la firma de un contrato (ya sea físico o virtual) que vincula las partes involucradas en el acuerdo. Las plataformas de mercado se pueden dividir en varias secciones en las que se formalizan acuerdos de forma independiente al resto del funcionamiento del mercado. Este concepto, al que se referirá posteriormente como mesa de negociación, es clave para formalizar acuerdos en una plataforma virtual de manera análoga a los acuerdos alcanzados en un entorno físico.

Dentro de una mesa de negociación las partes implicadas necesitan una manera de actuar común para poder alcanzar acuerdos de forma precisa y vinculante. Es por ello que se requiere definir un protocolo de negociación dentro de una mesa de negociación que permita estandarizar las acciones que se pueden realizar, su orden y sus posibles restricciones así como los detalles del contrato generado por el acuerdo que vincule a las partes involucradas. Ejemplos de estos protocolos de negociación son los correspondientes a las negociaciones entre dos o más personas de forma directa para la compra de productos alcanzando un acuerdo en el precio a pagar para las partes implicadas (protocolos *face to face*), las subastas en las que se pujan en base a unas reglas predefinidas y los protocolos de compra directa en la que el vendedor introduce un producto o conjunto a productos a vender y los compradores directamente compran las unidades o lotes que deseen adquirir (protocolos *direct trade*).

Por último, cabe destacar que con la necesidad de ajustar los comportamientos de los usuarios dentro de la plataforma y limitar sus actuaciones en base a la regulación interna de la misma resulta necesaria la aplicación de un contexto normativo consistente en un conjunto de normas estructuradas y categorizadas. La necesidad de establecer este contexto condiciona el uso de metodologías que contemplen este apartado.

## **2.2. Agentes software y sistemas multiagente**

En esta sección se introduce las tecnologías empleadas en el campo de los sistemas multiagente y las tecnologías del acuerdo. Para ello es necesario definir previamente el concepto de agente y su relación dentro de un sistema formado por varios agentes.

### **2.2.1. Definición de agente**

El uso de agentes software es el fundamento en el que se basa el procedimiento de automatización de procesos de compra y venta de productos y servicios en el mercado que se plantea desarrollar. En su definición clásica, y tal como define Michael Wooldridge [25], un agente software viene definido como una producción de software que dispone de características inherentes tales como persistencia (su ejecución se produce de forma continua), autonomía (capacidad propia de seleccionar, priorizar y tomar decisiones dentro del contexto en el que se sitúa el agente), proactividad (capacidad de reaccionar a los estímulos del entorno en el que se sitúa el agente) y capacidad social (capacidad de coordinación y/o colaboración con otras entidades del sistema).

### **2.2.2. Sistemas multiagente**

Un sistema en el que se agrupan diversos agentes se denomina sistema multiagente. En estos sistemas, los agentes disponen de un entorno en el que interaccionan conjuntamente tanto si los agentes disponen de objetivos personales propios (agentes auto-interesados) como si los agentes disponen de objetivos en común para favorecer a la sociedad a la que pertenecen (agentes benévolos). Es por ello por lo que en los

sistemas multiagente se agrupan en grupos denominados organizaciones. Las organizaciones de agentes suelen compartir algunos objetivos para una determinada entidad (por ejemplo, una empresa).

Estos sistemas también se pueden categorizar en función de quien controla los agentes en la que se distinguen los sistemas abiertos, en la que los agentes son controlados por varias organizaciones (empresas, equipos, etc.), o cerrados, en la que los agentes son controlados por una única organización.

En el contexto académico se han desarrollado diversas aplicaciones que emplean dichos sistemas multiagente, tales como su estudio desde la perspectiva del aprendizaje automático [23], desde la perspectiva lógica y de teoría de juegos [22] y desde la perspectiva de la Sociología [21]. En el contexto industrial se han aplicado los conceptos académicos en productos software tales como las orientadas al desarrollo de metodologías orientadas al software [26], la gestión de servicios de brokering [12] o la realización de estudios urbanísticos [19].

Un estudio más extenso de los sistemas multiagente se puede encontrar en [25] o en [7].

## 2.3. Tecnologías del acuerdo

El acuerdo es uno de los conceptos más importantes que permiten que los agentes humanos puedan interactuar con el entorno y es por ello que está representado en todas las facetas de la relación entre estos agentes y la sociedad. Este concepto permite la cooperación y la creación de sistemas sociales que permiten satisfacer necesidad en la vida.

En paralelo a su estudio en el campo de la Filosofía, la Sociología y su reciente aparición en campos de la Psicología y la Neurociencia se ha comenzado a estudiar en los últimos años en el contexto de los sistemas multiagente para su aplicación en tecnologías relacionadas con la sociedad de la información [21].

El acuerdo se define como la decisión tomada entre dos o más entidades en un contexto social determinado. Ejemplos de acuerdos en la sociedad se observan en las distintas subastas ([16],[11]), la compra de un producto en una plataforma de mercado de forma automática o la gestión de los derechos de agua en situaciones de escasez

([3],[5],[20]).

Un estudio más extenso de las tecnologías del acuerdo y sus dominios de aplicación se puede encontrar en [13].

### **2.3.1. Negociación automática**

Uno de las tecnologías del acuerdo más empleadas para el alcance de acuerdos en entornos competitivos y cooperativos es el uso de la negociación automática entre entidades o agentes. En la negociación automática dos o más agentes software alcanzan un acuerdo en base a la información de la que disponen sobre el contexto en el que se ubican, los objetivos tanto individuales como globales respecto al sistema multiagente al que pertenecen. La negociación automática en sus diversos entornos de aplicación ayuda mediante el uso de un protocolo común entre las partes agilizar el alcance de acuerdos aplicando la velocidad por la cual los computadores pueden procesar la información, permitiendo facilitar la negociación el múltiples acuerdos al mismo tiempo y alcanzar un acuerdo en menos tiempo. Estas dos capacidades permiten su uso en aplicaciones críticas tanto en la bolsa como en el salvamiento de personas atrapadas en accidentes.

Con el estudio de las tecnologías relacionadas con la negociación automática se da por finalizado un breve resumen del estado del arte de las tecnologías que engloban el trabajo y se procede al análisis y diseño del producto.

# Capítulo 3

## Análisis y diseño

En este capítulo comienza el análisis y diseño del trabajo. Para ello, en primer lugar, se describe un modelo de negocio que exponga las necesidades que se deberán estudiar para poder realizar el desarrollo del producto. Posteriormente, y en base a las conclusiones alcanzadas en el desarrollo del modelo de negocio, se emplea una metodología orientada a sistemas multiagente abiertos para definir la organización y estructura de las plataformas de mercado anteriormente descritas.

### 3.1. Modelo de negocio

Con el objetivo de validar el producto propuesto y orientarlo a su posterior comercialización se propone definir el marco en el cual se desarrollará el modelo de negocio del producto terminado. Esta etapa constituye el inicio de la etapa de análisis del trabajo a realizar y plantea encaminar el posterior trabajo a la producción de un producto final. Para ello se emplea el modelo *Business Model Canvas* [15]. Este modelo resume de forma rápida y fácil de entender por personas con un perfil no técnico los objetivos del producto a producir, su público objetivo, el tratamiento de los clientes, los canales de distribución, estructura de costes y fuentes de obtención de ingresos.

Formalmente el modelo *Business Model Canvas* estructura su información como sigue:

- **Propuesta de valor** (*Proposition value*), en la que se especifica cual es el producto y el objetivo que pretende cumplir de cara a los posibles clientes.



- **Segmentos de clientes** (*Customer segments*), en la que se especifica cuales son los distintos tipos de usuario de la plataforma o servicio a ofrecer y de los productos que se manejan en el.
- **Canales de distribución** (*Distribution channels*), en la que se especifican cuales son los canales en los que se pueden encontrar potenciales clientes y ofrecerles la propuesta de valor.
- **Relaciones con clientes** (*Customer Relationships*), en la que se especifican cuales son las principales acciones a realizar para mantener una relación de confianza con el cliente para poder ofrecer un servicio de acorde con la propuesta de valor.
- **Fuentes de ingresos** (*Revenue Streams*), en la que se especifican cuales son los recursos económicos generados mediante la venta u ofrecimiento de la propuesta de valor.
- **Recursos clave** (*Resources*), en la que se especifica los principales recursos necesarios (físicos y virtuales) para poder implementar el modelo de negocio propuesto.
- **Actividades clave** (*Key Activities*), en la que se especifican cuales son las principales actividades necesarias para poder implementar y posibilitar la relación con los potenciales clientes para ofrecer la propuesta de valor.
- **Socios clave** (*Key Partners*), en la que se especifican los principales interesados en el desarrollo del modelo de negocio y sus motivaciones para serlo.
- **Estructura de costes** (*Cost Structure*), en la que se especifican cuáles son los principales orígenes de costes en el modelo de negocio, su importancia y relacionar tanto las actividades como los recursos definidos anteriormente con sus costes.

Una definición más extensa de este modelo se puede encontrar en [14]. Aplicando el modelo a la descripción del problema y contexto mostrados en el capítulo de introducción, se crea un prototipo de modelo de negocio que se puede aplicar al producto que se pretende implementar. A continuación se detallan estas fases.

Un resumen del modelo generado se puede observar en la Figura 3.1.

### 3.1.1. Propuesta de valor

La propuesta de valor del producto consiste en ofrecer un servicio al cliente que permita la compraventa de productos y/o servicios de forma automatizada y autorregulada. Esto supone para el cliente una reducción sustancial del tiempo y recursos requeridos para la compraventa de estos productos ya que se facilitará la compra a precios competitivos y favorecedores de las partes involucradas y permitirá a los clientes de la plataforma disponer de más tiempo para desarrollar sus procesos de negocio y trabajar en el mantenimiento de éstos.

Por otra parte, su adaptabilidad a distintos dominios de aplicación permite a los clientes disponer de múltiples vías de vender productos que no habrían planteado su venta por el coste de la creación de un canal de distribución o los costes asociados en recursos humanos necesarios para establecer estos canales. Por ejemplo, una empresa dedicada a la venta de muebles puede además vender piezas propias específicas para otros sectores como la automoción (acabados en madera para algunos modelos de vehículo).

En resumen, las proposiciones de valor son las siguientes:

- Novedad
- Rendimiento
- Personalización
- Reducción de costes

### 3.1.2. Segmentos de cliente

Teniendo en cuenta los posibles dominios de aplicación a los que se puede aplicar el producto, se resumen los segmentos de cliente en tres tipos de clientes. En primer lugar destacan las empresas y agrupaciones empresariales (cooperativas, asociaciones, etc.) que desean a gran escala la compraventa de productos, y ofrecer y consumir servicios.

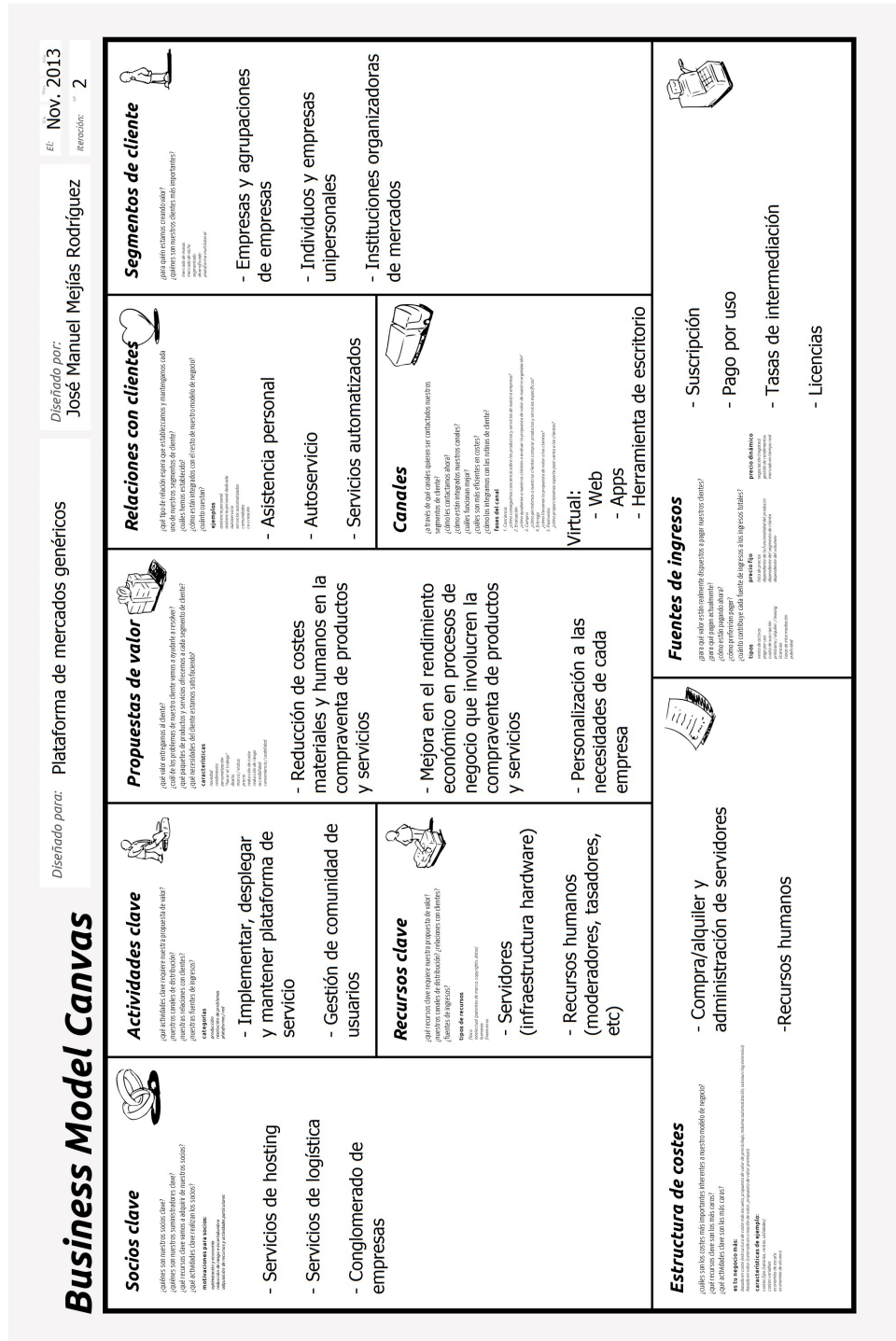


Figura 3.1: Business model canvas del proyecto

Por otra parte destacan los individuos particulares o empresas unipersonales (autónomos), que requieren a una menor escala las necesidades de las empresas y destinan una menor cantidad de recursos a los procesos de compra venta.

Por último forma parte de estos segmentos de cliente las agrupaciones que organizan los posibles mercados, cuyos intereses son de ofrecer una plataforma a sus clientes que pueden emplear para ofrecer servicios adicionales como la moderación de conflictos, servicios de tasación, etc.

En resumen, los segmentos de clientes son los siguientes:

- Empresas y agrupaciones de empresas
- Individuos y empresas unipersonales
- Instituciones organizadoras de mercados

### 3.1.3. Canales de distribución

Dada la naturaleza de la herramienta que se plantea desarrollar, el canal de distribución ha de ser virtual. Entre los diversos canales de distribución virtual disponibles destacan las plataformas web (de forma similar a otras plataformas de mercado disponibles en la red), las aplicaciones destinadas a dispositivos móviles (mercado emergente que puede complementar a las plataformas web) y las herramientas o aplicaciones de escritorio (orientadas a seguir el modelo de herramientas *ERP* existentes).

### 3.1.4. Relaciones con clientes

Para poder ofertar este tipo de soluciones informáticas es necesario establecer de forma directa relaciones con los potenciales clientes. Es por ello que se requiere que exista una asistencia personal que permita la transición desde los modelos tradicionales que emplean las empresas para la compraventa de productos y servicios en etapas iniciales del desarrollo del modelo de negocio. Por otra parte y conforme se pudiera escalar el número de clientes de la plataforma, será necesario establecer otros modelos como el autoservicio y servicios automatizados que

permitan de una forma sencilla establecer estas relaciones, así como disponer de un apoyo de una comunidad de usuarios.

### **3.1.5. Fuentes de ingresos**

La principal fuente de ingresos de la plataforma se basa en la suscripción a los servicios que se ofrecen en el mercado a las partes interesadas (principalmente a empresas y agrupaciones empresariales). Como fuentes de ingresos alternativas destacan el uso de tarifas para servicios concretos como tasaciones, moderación de conflictos, formalización y seguimiento de acuerdos de compraventa (orientado a individuos). Otro posible flujo de ingresos puede ser el uso de licencias para permitir a otras entidades el uso de las herramientas creadas (orientadas a los organizadores de mercados).

### **3.1.6. Recursos clave**

Los principales recursos claves son los asociados al mantenimiento de la plataforma de mercado. Por parte de la organización del mercado es necesario disponer de hardware específico para su mantenimiento incluyendo servidores que permitan gestionar un número elevado de peticiones simultaneas y de una infraestructura de red que permita ofrecer servicios a través de Internet.

En cuanto a los recursos humanos necesarios destacan los que son necesarios para ofrecer servicios dentro del mercado tales como moderadores, tasadores, etc.

### **3.1.7. Actividades clave**

Las principales actividades clave necesarias para desarrollar el modelo de negocio consisten en implementar, desplegar y mantener una plataforma que permita ofrecer los servicios clave de la propuesta de valor. Es por ello por lo que es necesario implementar un software especializado que permita crear plataformas de mercado y desarrollar las actividades de los distintos recursos humanos definidos

anteriormente. Por otra parte es conveniente el desarrollo de una herramienta auxiliar que permita gestionar la comunidad de usuarios.

### **3.1.8. Socios clave**

Los principales socios clave para el desarrollo de la idea de negocio son los suministradores de servicios de hosting de servidores que permitan desplegar la infraestructura requerida para el desarrollo de las plataformas de mercado. Además dependiendo de la naturaleza del mercado en cuestión, se plantean otros socios clave como son empresas de gestión logística, conglomerados de empresas (agrupaciones de comercio, cooperativas, etc) y asociaciones de consumidores.

### **3.1.9. Estructura de costes**

Los costes del proyecto se dividen en dos. En primer lugar destacan los costes relacionados con la infraestructura (servidores y el mantenimiento de los mismos). Los segundos son los relacionados con los recursos humanos necesarios para mantener el servicio.

## **3.2. Análisis y diseño del sistema mediante la metodología ROMAS**

En este capítulo se la metodología permite analizar los requisitos y diseñar el sistema a implementar. La metodología permite explicar el sistema diseñado desde un punto de vista general hasta un esquema detallado de las interacciones que se observan en el sistema y de los actores que lo realizan (agrupados por roles).

### **3.2.1. Metodología de diseño (ROMAS)**

En esta sección se expone al lector la metodología empleada.

## Introducción

Este proyecto usa como base para su diseño e implementación la metodología de modelado llamada ROMAS (*Regulated Open MultiAgent Systems* o Sistemas Multi-Agente Regulados y Abiertos). ROMAS se compone de un metamodelo específico, una metodología que guía el proceso y una herramienta CASE de modelado y generación automática de código. El metamodelo y la metodología fue definido siguiendo el estándar FIPA *Design Process Documentation Template* [6].

ROMAS permite especificar sistemas normativos abiertos como sistemas multi-agente que constituyen una organización, en donde los agentes interactúan mediante el uso de servicios y donde los acuerdos alcanzados se formalizan mediante normas y contratos. La metodología ROMAS basa sus producciones en la creación de una arquitectura orientada a servicios. Las organizaciones en esta arquitectura representan un conjunto de personas e instituciones que tienen que coordinar recursos y servicios. Las partes individuales de estas organizaciones son los agentes. Estos agentes asumen roles dentro del sistema (y por ende, de la organización), pueden ofrecer y consumir servicios como parte de las funciones que desempeñan. Por otra parte, las organizaciones también pueden ser creadas para coordinar recursos y servicios a través de límites institucionales. Las normas determinan los permisos, obligaciones y prohibiciones que restringen el comportamiento de los agentes del sistema. Los contratos se utilizan para formalizar los acuerdos entre las entidades.

El metamodelo de ROMAS consta de 4 vistas:

- *Vista organizacional*: se especifican los objetivos globales, la funcionalidad ofrecida al entorno y la estructura social de la organización.
- *Vista interna*: se especifica la funcionalidad interna, capacidades, creencias y objetivos de cada entidad del sistema.
- *Vista de la plantilla del contrato*: se especifican las restricciones predefinidas que todo contrato de un determinado tipo debe cumplir.
- *Vista de la actividad*: se especifican los protocolos de interacción y la secuencia de actividades en la que las tareas y servicios se descomponen.

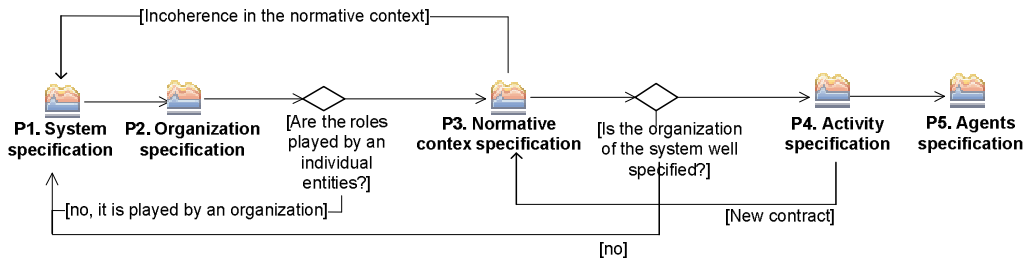


Figura 3.2: Diagrama ilustrativo de las fases de la metodología

El proceso de desarrollo de la metodología está compuesto por 5 fases que se muestran en la Figura 3.2 y se detallan a continuación:

1. *Fase 1: Especificación del sistema:* se analizan los requisitos del sistema desde un punto de vista global, se definen los objetivos que se desean cumplir de forma general, se subdividen estos objetivos en subobjetivos y se define una serie de tareas en las que se indicarán las acciones que se deberán realizar para cumplir dichos objetivos. Por último se especifican el orden de estas tareas que indicarán esquemas de actuación.
2. *Fase 2: Especificación de la organización:* se analiza y diseña la estructura social y funcional del sistema. En esta fase se identifican los roles que interactúan en el sistema. Posteriormente se relacionan los roles definidos con las tareas definidas en la Fase 1 e identifican las interacciones entre los distintos roles en algunas tareas, que definirán posteriormente protocolos de interacción.
3. *Fase 3: Especificación del contexto normativo:* se identifican y formalizan las normas y contratos del sistema. En base a la organización formulada y sus roles se especifica cuales son las restricciones que deben de existir para controlar las acciones que los agentes que formarán el sistema deberán respetar.
4. *Fase 4: Especificación de las actividades:* se detallan los protocolos de interacción en base a las interacciones definidas en la Fase 2 y la normativa generada anteriormente.
5. *Fase 5: Especificación de los agentes:* se analizan y diseñan las entidades individuales del sistema



### 3.2.2. Análisis y diseño del sistema

#### Fase 1: Especificación del sistema

En esta fase se analizan desde un punto de vista general y global los requisitos del sistema. Estos requisitos se traducen en objetivos y restricciones. Los objetivos globales del sistema se descomponen para determinar objetivos operacionales y casos de uso necesarios para la implementación del sistema. Finalmente, se analiza la adecuación de la metodología al problema propuesto.

Como resultados se obtiene:

1. *Descripción general del sistema*: se especifica desde un alto nivel la funcionalidad del sistema (Epígrafe 3.2.2.1).
2. *Desglose de objetivos*: una descomposición de objetivos en base a la descripción y objetivos generales (Epígrafe 3.2.2.2).
3. *Descripción general de tareas*: una descomposición de objetivos en tareas que permitan definir las unidades de las acciones que se tendrán que hacer en el sistema para cumplir los objetivos (Epígrafe 3.2.2.3).
4. *Diagramas de secuenciación de tareas*: se estructuran las tareas definidas anteriormente para determinar relaciones de precedencia, consecuencia y paralelismo de tareas (Epígrafe 3.2.2.4).
5. *Adecuación del sistema a la metodología ROMAS*: un estudio de la adecuación de la metodología, de los sistemas multiagente, de el uso organizaciones y el control normativo en sí al sistema propuesto (Epígrafe 3.2.2.5).

A continuación se muestran los documentos generados en esta fase.

#### 3.2.2.1 Descripción general del sistema

En la descripción del sistema, mostrado en el Cuadro 3.1 se muestra al usuario una descripción general y bajo un punto de vista general la funcionalidad del sistema, los objetivos a cumplir para su creación, los recursos necesarios para su uso y los servicios o funcionalidades que provee a sus usuarios.

Identificador del sistema	MAGEMA, acrónimo referido a MAgentix2 framework for the generation and automatic interaction with GEneric ontology MArkets
---------------------------	--

Descripción del sistema	El framework permite la creación de diversas plataformas de mercado en la que se pueden definir un conjunto de productos, servicios y elegir los protocolos de negociación en los que se puede negociar en la plataforma que hayan sido definidos previamente. El sistema ha de adaptarse a casos donde un alto número de agentes está presente en la plataforma por lo que se aconseja el empleo de una arquitectura descentralizada. Su objetivo es facilitar las interacciones comerciales entre las entidades del sistema mediante su automatización mediante el uso de agentes controlables por los usuarios de la plataforma. El sistema ha de adaptarse a casos donde un alto número de agentes está presente en la plataforma por lo que se aconseja el empleo de una arquitectura descentralizada. Se debe asegurar su flexibilidad a diversos tipos de protocolos conversacionales y tipos de producto.
Dominio	El sistema se plantea para su uso en un dominio genérico en el que se requiera negociaciones para llegar a acuerdos mediante el intercambio de bienes entre las entidades del sistema.
Tipo de entorno	El entorno en el que se sitúa el sistema se asemeja al que un mercado físico se posicionaría en el mundo. En ella se definen roles característicos del dominio donde se aplica el sistema distribuido entre distintas entidades a las que se le pueden asignar recursos de la plataforma y/o manejadores de eventos ocurridos durante su funcionamiento. La interacción con el usuario en el sistema es un pilar de la plataforma y forma parte de todos los aspectos del mercado en las que se puede actuar en la plataforma (desde la creación del mercado hasta su cese de actividad, pasando por la interacción con los agentes creados). Para facilitar dicha interacción se considera necesario el uso de un entorno gráfico.
Objetivos globales	El sistema debe permitir la creación de plataformas de mercado en base a los objetivos de la empresa y de los protocolos que se consideren apropiados para la venta de productos y servicios de la plataforma. El sistema debe permitir interactuar con los agentes de un mercado ya formado facilitando las labores de compra y venta de productos además de facilitar servicios de listado y anuncio de los mismos. El sistema ha de conseguir reducir el esfuerzo de las partes del mercado en alcanzar acuerdos de intercambio de bienes empleando una interfaz amigable y sencilla de usar ajustada a los requisitos de ejecutivos de empresas y, a su vez, a los requisitos de gente inexperta en el uso de computadores y mercados. Por otra parte el sistema debe proporcionar ayuda a administradores de sistemas informáticos y técnicos en general a la instalación y mantenimiento del sistema de una forma sencilla y adaptable a las necesidades de la empresa a la que se distribuya el mismo. Finalmente, el sistema debe permitir a administradores, asesores, contables y gerentes (entre otros) acceder a la información acerca de las transacciones realizadas en el mercado, gestionando su ciclo de actividad y recabando información de forma estructurada para la posterior realización de informes y análisis estadísticos.
Recursos	Base de datos: recurso necesario para registrar las operaciones del mercado, almacenar las credenciales de acceso de los usuarios de la plataforma y como base para asegurar la persistencia del sistema.

Ofertas	El sistema ofrece a sus usuarios un creador de plataformas en la que se pueden crear plataformas de mercado configurable acorde a las necesidades de los usuarios que conforman las organizaciones del mercado. Para los proveedores se les ofrece la plataforma como medio para listar sus productos y alcanzar acuerdos sobre estos productos para su venta. En cuanto a los consumidores, la plataforma ofrece servicios para la compra automatizada de productos, la interacción directa con el mercado para dichas compras si lo desean así como listar las necesidades en productos como punto de partida para alcanzar acuerdos.
---------	---

Cuadro 3.1: Descripción general del sistema propuesto

### 3.2.2.2 Desglose de objetivos

A partir de la descripción del sistema se extraen los objetivos globales y se descomponen, especifican y jerarquizan los objetivos del sistema en un diagrama de objetivos como el que se puede observar en la Figura 3.3. En este diagrama, se parten de los objetivos generales especificados en el documento general del sistema (Cuadro 3.1) y se detallan los objetivos mediante una jerarquía descendente en la cual los objetivos (tanto abstractos, o generales, como operacionales, específicos) se descomponen en subobjetivos hasta alcanzar objetivos muy específicos (puramente operacionales) de las cuales se especificarán tareas y restricciones necesarias para su cumplimiento.

Podemos observar a partir de los objetivos generales del sistema surgen cuatro objetivos que son la definición y lanzamiento del mercado, la monitorización y uso del mercado y por último el fomento de la reutilización de la estructura del mercado. A partir del objetivo de definición del mercado se definen objetivos operacionales que tratan tanto las fases que el usuario deberá seguir para definir el mercado así como su lanzamiento.

En la parte de monitorización de mercado se engloban objetivos que tratan las diversas funcionalidades a implementar para poder asegurar el correcto funcionamiento del mercado. Se incluye de partida el control de credenciales para poder identificar a los usuarios del sistema dentro de la plataforma ya creada, además de la administración de los usuarios en la cual se gestionarán los registros manuales, las peticiones de registro, los datos de los usuarios, los permisos y sanciones y las diversas listas de potenciales usuarios de la plataforma para labores externas. En cuanto a la gestión del propio mercado se incluye la actualización de la especificación del mercado durante el funcionamiento del mercado de la misma forma que se ha definido el mercado previamente, el control del ciclo de vida del mercado, visualizar estadísticas del mer-

cado, visualizar el estado del mercado en tiempo real, moderar los posibles conflictos, gestionar las mesas de negociación y tasar productos.

La interacción con el mercado incluye la gestión por parte del usuario de sus datos, y realizar procesos de compraventa de productos además de gestionar el funcionamiento del agente autónomo que lo representa. La compraventa incluye la visualización del mercado, incluyendo las mesas de negociación abiertas, la negociación dentro de la mesa de negociación en base a un protocolo dado para la mesa, el alcance y formulación de acuerdos, la solicitud de la apertura de una mesa de negociación, la solicitud de un tasado del producto y la visualización y gestión de los acuerdos alcanzados por el usuario. Finalmente, la gestión del usuario autónomo incluye el control del ciclo de vida del mismo y la especificación de su funcionamiento alterando parámetros como sus objetivos de compraventa, su función de utilidad o el presupuesto del que dispone.

### **3.2.2.3 Descripción general de tareas**

A continuación se analiza como se van a conseguir esos objetivos asociándolas a tareas. Para ello detallan los objetivos operacionales anteriormente fijados y se obtienen diagramas de tareas asociados a los objetivos de a las que a su vez se pueden definir subtareas. Dado que se tratarán las tareas definidas y se secuenciarán en el siguiente epígrafe la generación de taras se detallará en el Apéndice A.

### **3.2.2.4 Diagramas de secuenciación de tareas**

A continuación se definen los diagramas de secuencia, en los cuales usando las tareas definidas en el epígrafe anterior se especifica un orden en el cual las tareas se relacionan entre ellas a través del tiempo y del flujo de interacción. El objetivo es obtener una secuencia de acciones o tareas que se deben cumplir secuencialmente o en paralelo. En primer lugar se puede observar en la Figura 3.4 el flujo de interacción de las tareas desde un punto de vista general, el cual se extiende en las siguientes figuras para dar un enfoque más detallado al flujo de interacción de las diferentes partes del sistema. Se puede observar en la figura como hay una paralelización entre la identificación de usuarios, el registro de los usuarios y recordar la contraseña. Posteriormente a este proceso de entrada se toma un eje de acción en el que dependiendo del rol de la entidad identificada. Este eje de acción se divide en tres aspectos que se detallan en las figuras posteriores.

En la Figura 3.5 se puede observar la secuencia de tareas que envuelven la gestión

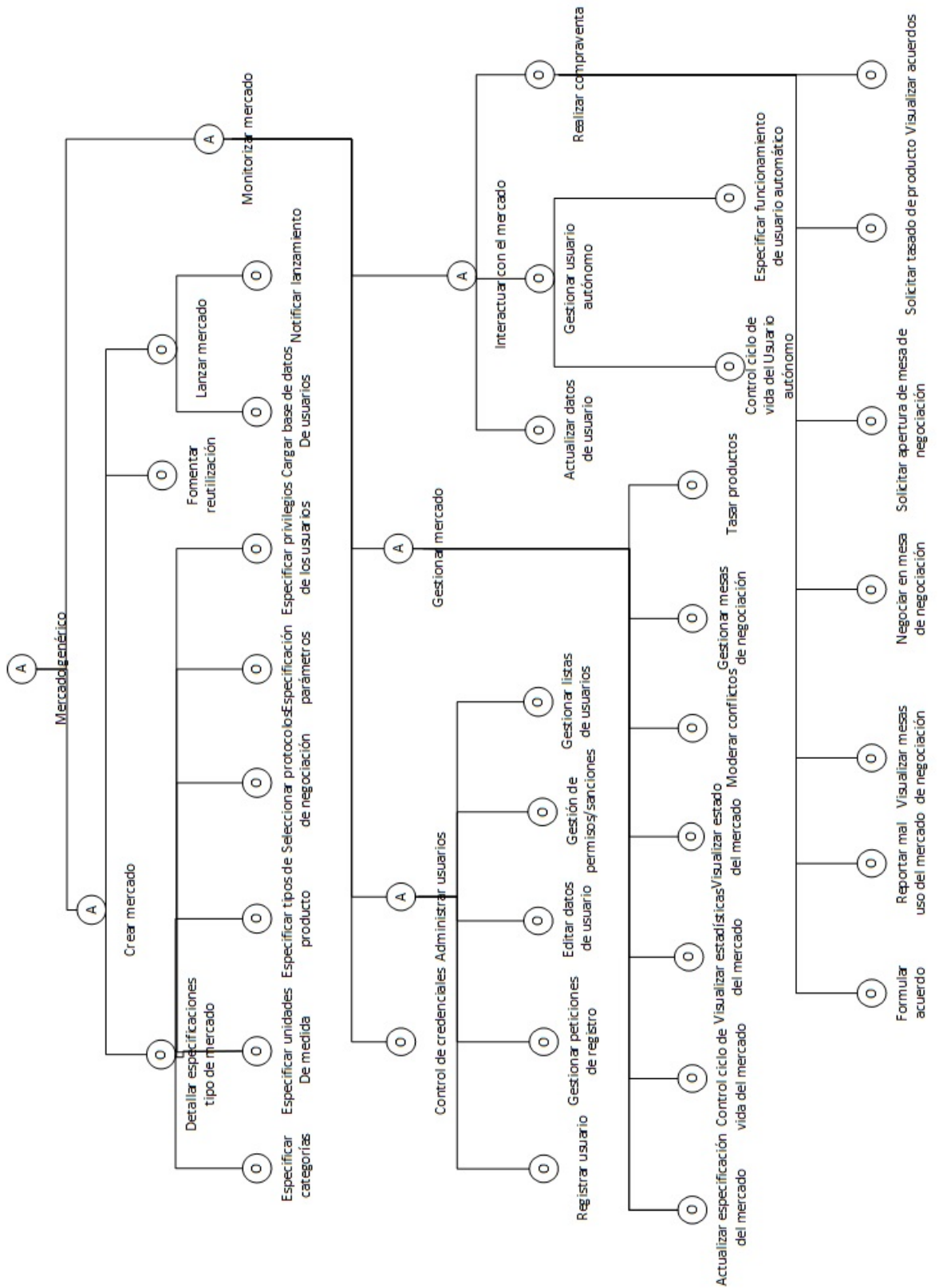


Figura 3.3: Diagrama de objetivos del sistema

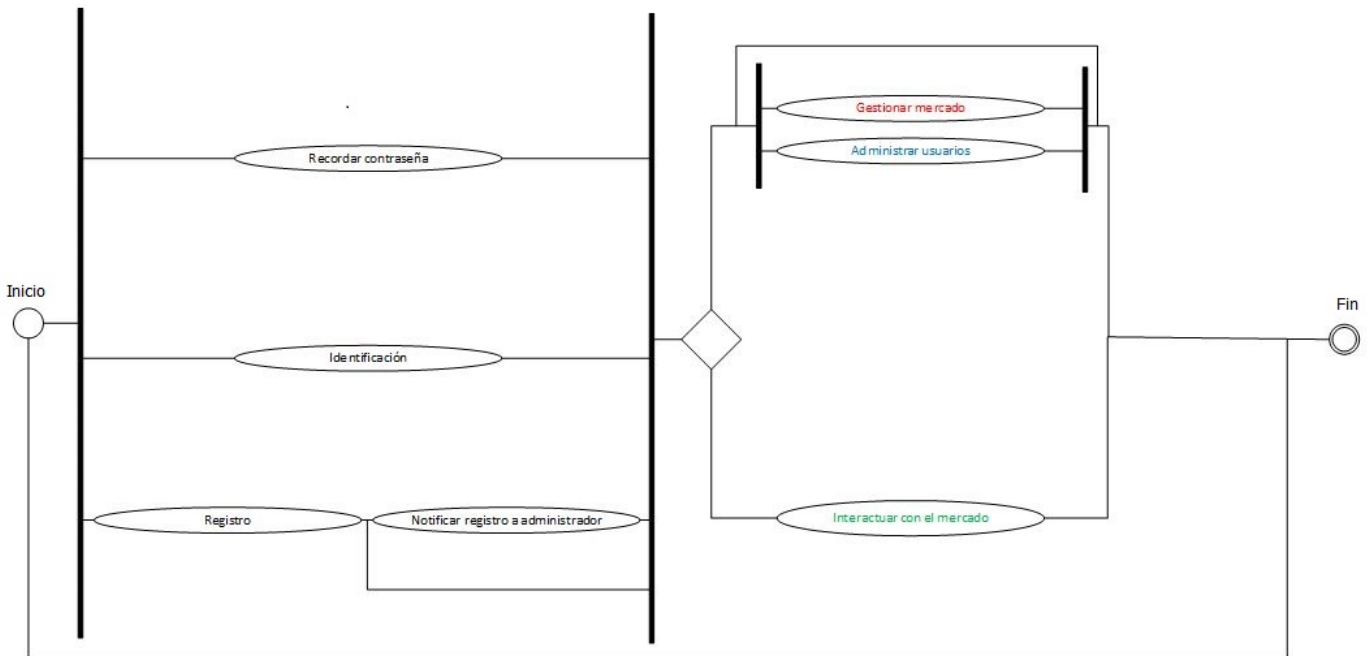


Figura 3.4: Diagrama de secuencia desde un punto de vista general

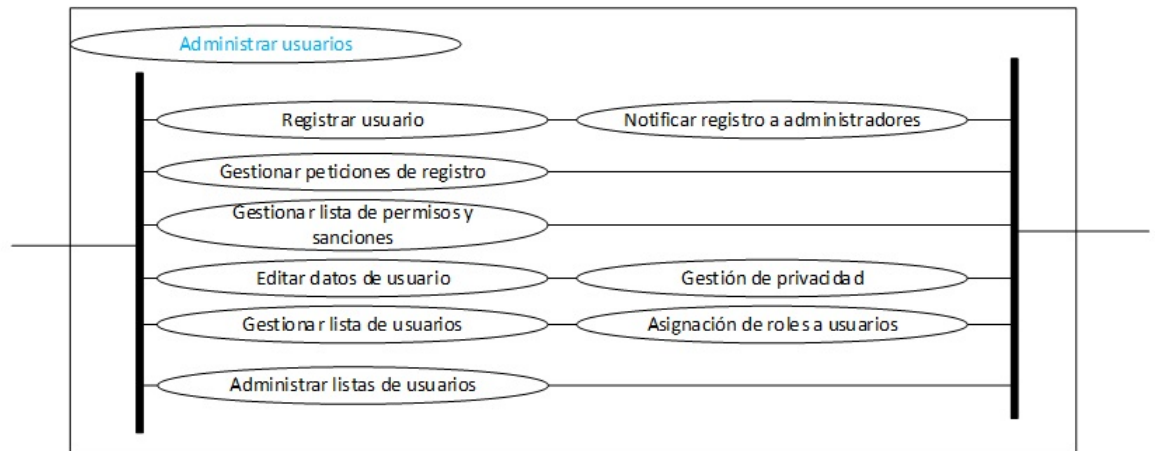


Figura 3.5: Diagrama de secuencia orientado a la gestión de usuarios de la plataforma de los usuarios de la plataforma y sus roles dentro de ella. La tarea *Administrar usuarios* definida en la Figura 3.4 se compone por las siguientes tareas que se secuencian en paralelo:

- Registro de usuarios de forma manual, incluyendo su posterior notificación de registro a administradores del mercado.
- Gestionar las peticiones de registro de usuarios dentro de la plataforma.
- Gestionar la lista de permisos y sanciones, permitiendo controlar las sanciones efectuadas de los usuarios registrados dentro de la plataforma.
- Editar los datos de los usuarios del mercado de forma manual, incluyendo la gestión de la privacidad de los usuarios de la plataforma.
- Gestionar la lista de usuarios registrados en la plataforma y la asignación de roles adicionales o revocación de los mismos.
- Administración de listas de usuarios registrados, orientada a la gestión de los usuarios registrados para su exportación o importación a otras plataformas de mercado.

En la Figura 3.6 se puede observar la secuencia de tareas que envuelven la gestión del mercado en sí. Entre las tareas que se pueden realizar secuencialmente se incluyen:

- Actualizar la especificación del mercado mientras la plataforma de mercado se ejecuta.
- Controlar el ciclo de vida del mercado, permitiendo pausar temporalmente, cesar permanentemente la actividad del mercado o reanudar su actividad tras una anterior pausa.
- Visualizar el estado del mercado y las estadísticas más detalladas del mercado.
- Gestionar la lista de conflictos generados en la plataforma entre los usuarios, permitiendo resolver los conflictos de forma manual.
- Gestionar las peticiones de sanción ocurridas por la resolución de algunos de los conflictos ocurridos dentro de la plataforma o por la violación de las normas de la misma.
- Gestionar la lista de peticiones de tasación de los usuarios de la plataforma, que posteriormente se pueden resolver mediante una tasación automática o crear una norma para que la tasación se realice de forma automática.
- Formalizar de forma manual acuerdos dentro de la plataforma.

Por último, la interacción con el mercado de los usuarios de la plataforma se muestra en la Figura 3.7. Esta interacción se divide en las siguientes tareas en paralelo:

- Listar las mesas de negociación que forman parte de la plataforma, para posteriormente acceder y ver los detalles de la mesa de negociación y finalmente negociar dentro de la mesa de negociación para alcanzar un posible acuerdo.
- Ver la información de un usuario de la plataforma, y reportar el usuario si ha violado las normas de la plataforma.
- Realizar una petición de tasación de un producto o servicio determinado
- Realizar una petición de apertura de una mesa de negociación, incluyendo la definición del producto o servicio con el que se desea negociar.



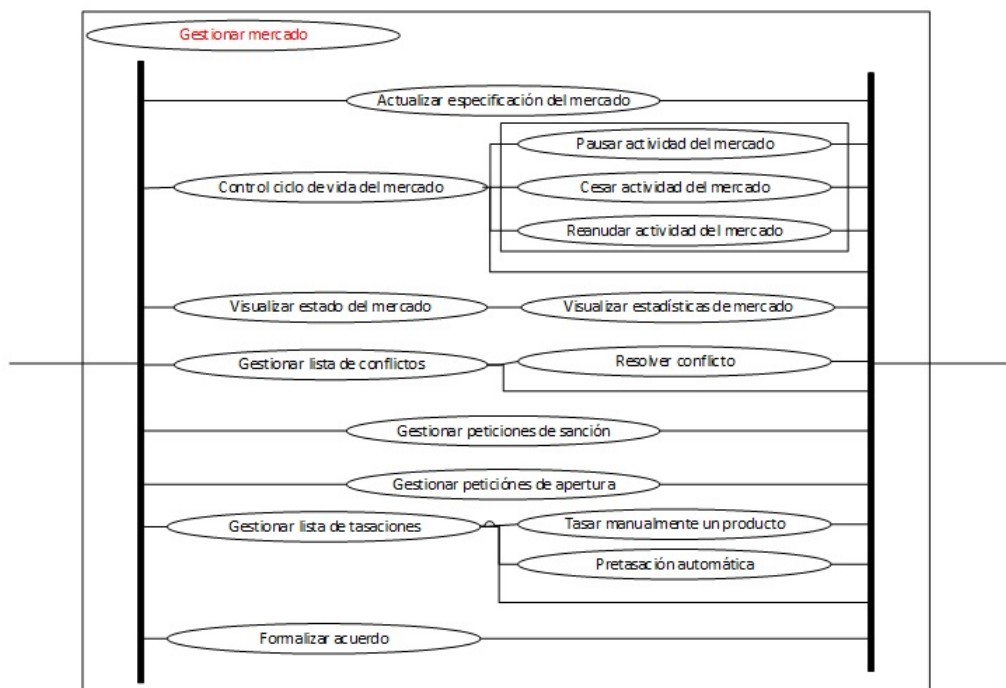


Figura 3.6: Diagrama de secuencia orientada a la gestión del mercado

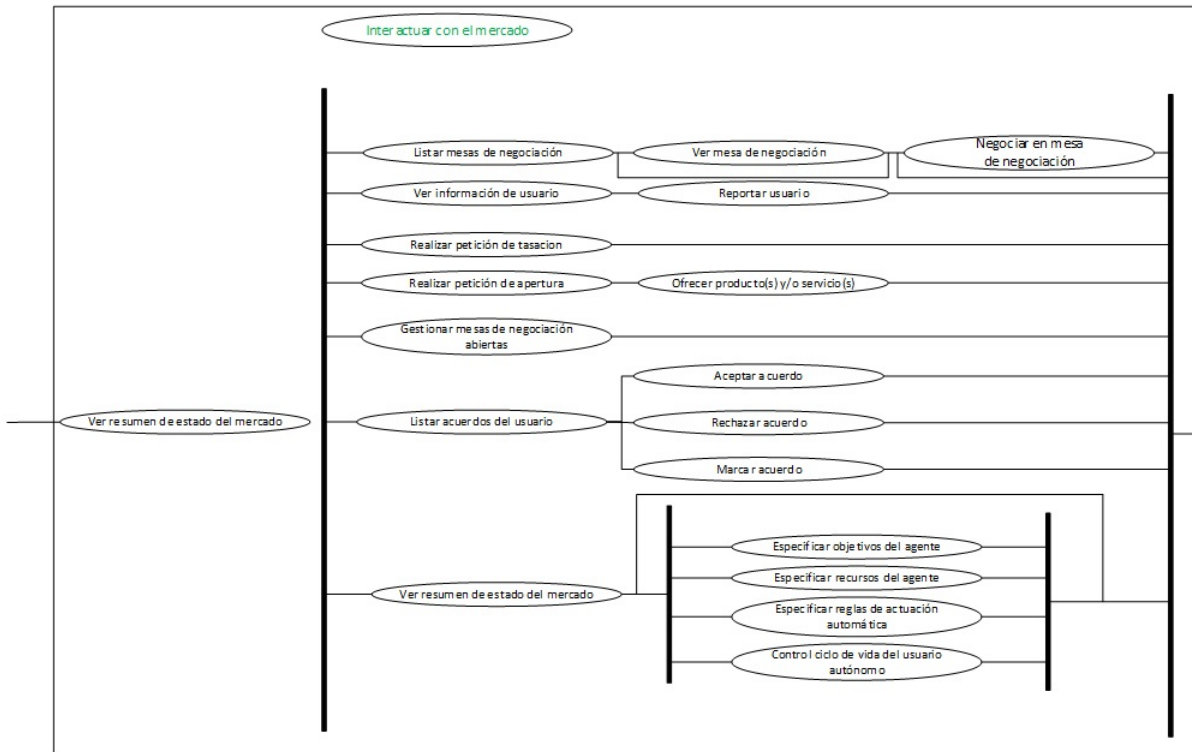


Figura 3.7: Diagrama de secuencia orientado a la interacción con el mercado

- Gestionar las mesas de negociación abiertas por un usuario de la plataforma.
- Listar los acuerdos de un usuario de la plataforma incluyendo la posibilidad de aceptar, rechazar o marcar una propiedad o parte del acuerdo.
- Ver un resumen del estado del mercado y del agente que lo representa, al cual puede especificar sus objetivos (vender/comprar productos, ofrecer/solicitar servicios), especificar los recursos (productos o divisas) del que dispone el agente, definir reglas para actuar dentro de una mesa de negociación y finalmente controlar su ciclo de vida.

### 3.2.2.5 Adecuación del sistema a la metodología ROMAS

La metodología ROMAS analiza en este punto tras el análisis genérico la viabilidad de diseñar la aplicación mediante los conceptos de agentes, roles, organizaciones, contratos, contextos normativos, entre otros.

Para ello se realizó un estudio en base a la especificación y documentación generada en esta fase y siguiendo las guías que se definen en la metodología para determinar si el sistema descrito es adecuado para poder ser desarrollado empleando tanto la metodología como el uso de agentes inteligentes como paradigma, su agrupación en sistemas multiagente, el uso de organizaciones dentro del sistema y el control normativo. Como resultado se concluye lo siguiente:

1. Se recomienda el uso de un sistema distribuido para la implementación del sistema debido a que el sistema se puede subdividir en subsistemas y el sistema está formado por diferentes entidades que interactúan entre ellos para cumplir objetivos globales.
2. Se recomienda el uso de agentes para la implementación del sistema ya que se presentan distintas entidades con objetivos distintos (los objetivos de los distintos clientes de las plataformas creadas, por ejemplo), las entidades del sistema son heterogéneas y se observan propiedades de proactividad y adaptabilidad.
3. Se recomienda el uso de soporte para organizaciones en el sistema dado la división de las tareas en diversos departamentos, el uso de una regulación dada por las restricciones del sistema y existe una jerarquía de usuarios equivalente a la existente en los sistemas reales a los que se imita.
4. Por último, se recomienda el uso de regulaciones en el sistema dada la naturaleza abierta del sistema, la existencia de entidades autointeresadas y la falta de control de todas las partes envueltas en el sistema.

En este punto finaliza la fase 1 de la metodología, en la cual se han analizado los aspectos generales del sistema y comprobado la adecuación del sistema y su naturaleza a la metodología a emplear. Posteriormente se especificará la estructura de la organización que controlará la ejecución del sistema.

## **Fase 2: Especificación de la organización**

En esta fase se analiza y diseña la estructura social del sistema. Las distintas funcionalidades del sistema se asocian a roles para posteriormente analizar las rela-

ciones entre los mismos, sus restricciones y el entorno en el que se desarrollan sus actividades.

Como resultado de aplicar la metodología en esta fase se obtuvo lo siguiente:

1. Definición y desglose de roles: se definen los roles que se emplearán en el sistema para cumplir las tareas definidas en la fase anterior, junto a su descripción (Epígrafe 3.2.2.1).
2. Organigrama: se reúnen los roles definidos y se especifica qué jerarquía hereditaria hay en la organización que representa el sistema además de las relaciones de autoridad entre los distintos roles (Epígrafe 3.2.2.2).
3. Diagrama de interacción: resume la interacción entre los distintos roles para determinar las tareas en las que hay dos o más roles asociados y requerirán desarrollar un protocolo de interacción (Epígrafe 3.2.2.3).

### 3.2.2.1 Definición y desglose de roles

En base a las tareas definidas en la primera fase y siguiendo la guía de identificación de roles de ROMAS se definen los roles que identifican los actores que interactúan con el sistema. Estos roles más adelante podrán ser tratados mediante agentes si se considera necesario. Los roles que se han obtenido en base a la metodología son los siguientes:

- *Designer* (Diseñador): este rol se encarga de la definición y lanzamiento de plataformas de mercado, gestionando toda la definición del mercado durante la etapa de diseño y programando la creación y lanzamiento de las plataformas de mercado en una fecha dada. Un diagrama resumen de su funcionalidad se puede observar en la Figura 3.8. En él podemos observar que el diseñador se encarga de gestionar las tareas relacionadas con la especificación de tipos de producto, las unidades de medida o categorías, entre otros. Estas tareas cumplen además con los objetivos de especificar categorías, tipos de producto, unidades de medida, etc. Además, este rol requiere como recurso una interfaz gráfica que permita diseñar el sistema ya sea mediante terminal o con una interfaz interactiva.

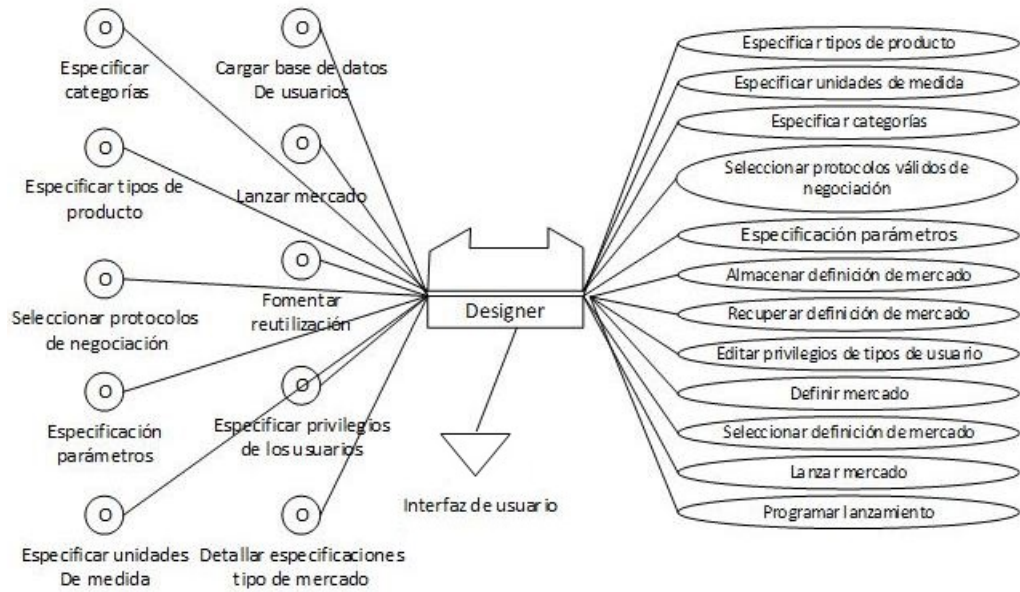


Figura 3.8: Diagrama que resume la funcionalidad del diseñador de la plataforma de mercado

- *Unregistered* (Usuario no registrado): este rol engloba a los usuarios de la plataforma de mercado ya creada que no se han identificado o registrado en el sistema. Sus acciones se relacionan con la acreditación de las credenciales necesarias para la identificación dentro del sistema ya sea mediante registro o identificación. Su funcionalidad se resume en la Figura 3.9. Como se puede ver, el rol correspondiente a un usuario no registrado solo cumple los objetivos control de credenciales y registrar usuario.

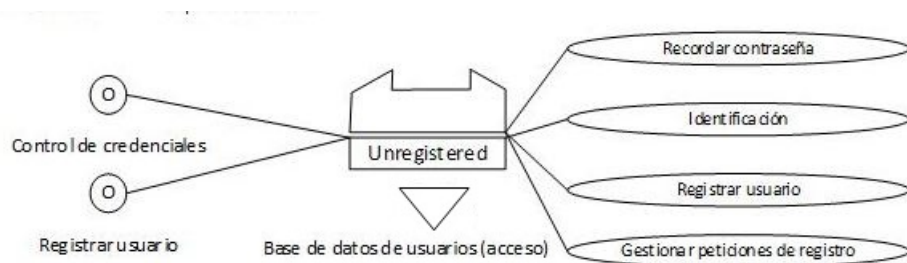


Figura 3.9: Diagrama que resume la funcionalidad del usuario no registrado en la plataforma del mercado

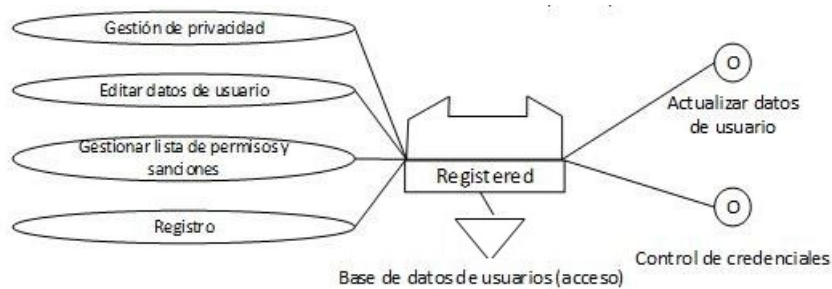


Figura 3.10: Diagrama que resume la funcionalidad del usuario registrado en la plataforma del mercado

- *Registered* (Usuario registrado): este rol engloba a los usuarios de la plataforma de mercado ya registrados e identificados. Sus acciones se engloban en la administración de las credenciales propias del usuario. Su diagrama resumen corresponde a la Figura 3.10. En ella se puede observar que el rol de usuario registrado como tal solo puede gestionar sus datos de usuario y controlar su registro dentro de la plataforma.
- *Market Manager* (Gestor del mercado): este rol engloba al encargado de gestionar el correcto funcionamiento del mercado. Sus acciones se engloban en la administración de usuarios de la plataforma de mercado, la administración del ciclo de vida de la misma y la visualización de su estado. Su funcionalidad viene dada en el diagrama resumen de la Figura 3.11. Cabe destacar la gran cantidad de tareas asignadas en este rol y es por ello por el que su actividad podrá ser distribuida entre varios agentes.
- *Rater* (Tasador): este rol engloba al encargado de gestionar las peticiones de tasación del sistema. Sus acciones se corresponden a las equivalentes a un tasador en la vida real. Se puede observar su funcionalidad como rol en la Figura 3.12 en el que gestiona el objetivo de tasar productos y sus tareas correspondientes.
- *Moderator*: este rol gestiona la moderación de conflictos dentro de la plataforma. Sus acciones se relacionana con la resolución de los conflictos acarrecidos en la plataforma y la posible petición de la suspensión de usuarios en la resolución de los mismos. Un resumen de las acciones de este rol se observa en la Figura

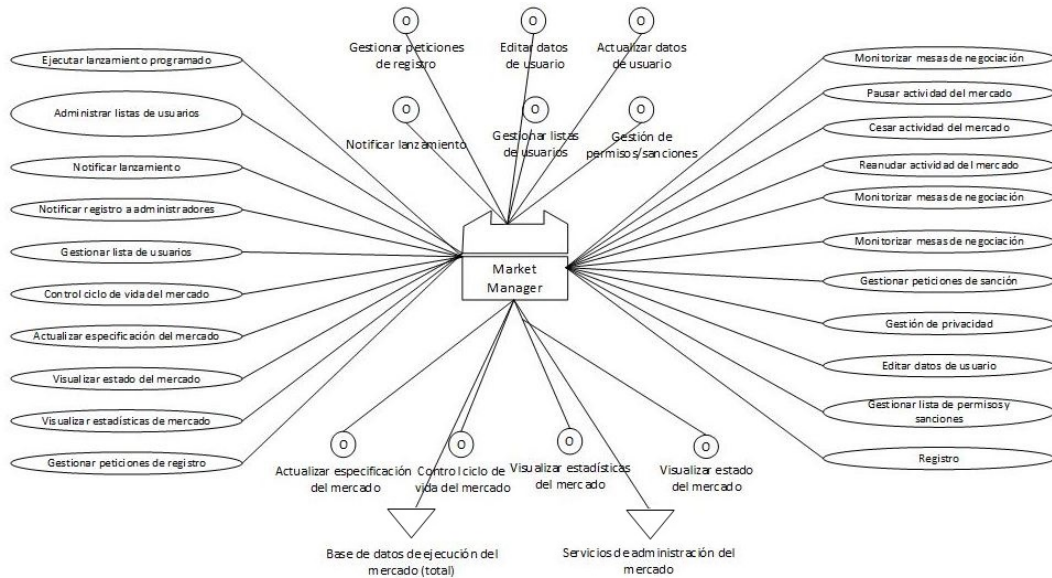


Figura 3.11: Diagrama que resume la funcionalidad del gestor de la plataforma de mercado

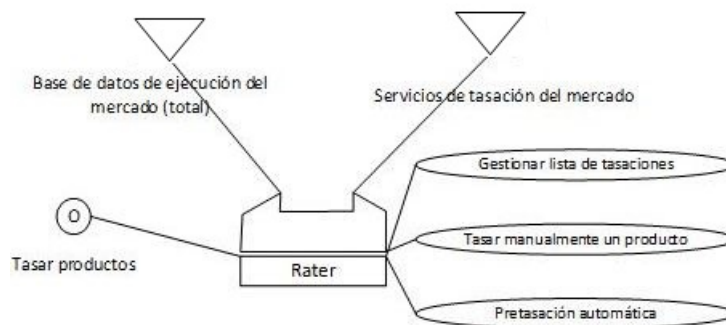


Figura 3.12: Diagrama que resume la funcionalidad del tasador de la plataforma de mercado

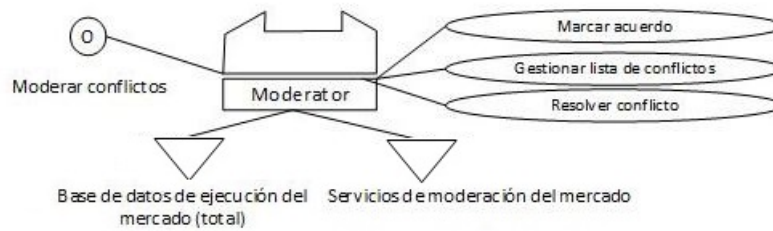


Figura 3.13: Diagrama que resume la funcionalidad del moderador de la plataforma de mercado

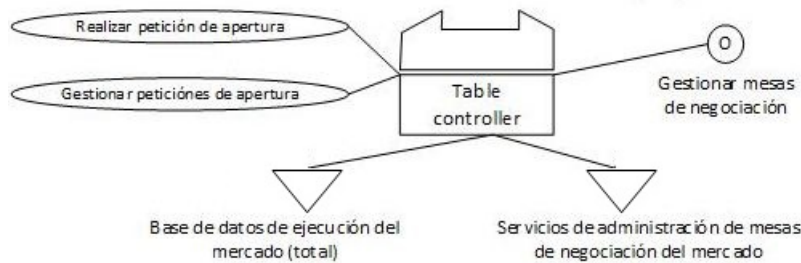


Figura 3.14: Diagrama que resume la funcionalidad del controlador de mesas de negociación de la plataforma de mercado

3.13. Sus acciones se ven limitadas ya que el gestor de mercado se encarga de la gestión de los usuarios y de controlar las sanciones de los mismos.

- *Table Controler*: este rol gestiona la revisión de las peticiones de apertura de mesas de negociación dentro de la plataforma. Un resumen de su funcionalidad se observa en la Figura 3.14. Su labor única consiste en satisfacer el objetivo correspondiente a la gestión de la apertura de mesas de la plataforma.
- *Table Judge*: este rol gestiona el funcionamiento dentro de las mesas de negociación en base a un protocolo dado. Sus labores incluyen la gestión de la mesa de negociación asegurando el correcto funcionamiento del mismo y la formalización de los acuerdos alcanzados. Un resumen de su funcionalidad se observa en la Figura 3.15.
- *Market user*: este rol gestiona la interacción con el mercado de los usuarios para la compraventa de productos. Sus acciones incluyen la visualización del estado



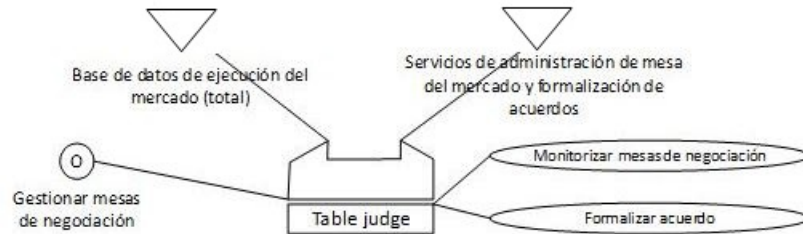


Figura 3.15: Diagrama que resume la funcionalidad del juez de mesa de negociación de la plataforma de mercado

del mercado, la negociación de acuerdos, la solicitud de apertura de mesas de negociación y tasación de productos, la revisión de los acuerdos dados y la gestión del agente autónomo que lo representa. Un resumen de su funcionalidad se observa en la Figura 3.16.

### 3.2.2.2 Organigrama

Posteriormente a la identificación de roles y definir sus interacciones con el sistema se define cómo se estructura la estructura social organizacional en la que están envueltos los roles que servirá posteriormente para realizar el control normativo y diseñar las operaciones e interacciones dentro del sistema. Como se observa en la Figura 3.17, la organización principal contiene un rol diseñador cuya función, como se ha explicado antes, es la de definir y lanzar a ejecución plataformas de mercado. Seguidamente se define la organización que engloba todos los roles que desarrollan la actividad dentro del a plataforma de mercado en ejecución.

En dicha organización se hace una división entre los roles representando a un usuario no registrado (*Unregistered*) y el que representa a un usuario registrado (*Registered*). En el caso de *Registered* se engloban todos los roles en los que se incluye tanto a la organización de la plataforma de mercado (*Staff*) como a los usuarios que realizan la compraventa en el mercado (*Market Users*). En el caso de la organización de mercado, dicho rol se extiende en los roles que representan al gestor del mercado (*Market Manager*), al moderador (*Moderator*), al tasador (*Rater*), al controlador de mesas (*Table Manager*) y finalmente al juez de mesa (*Table Judge*). Por último, el rol

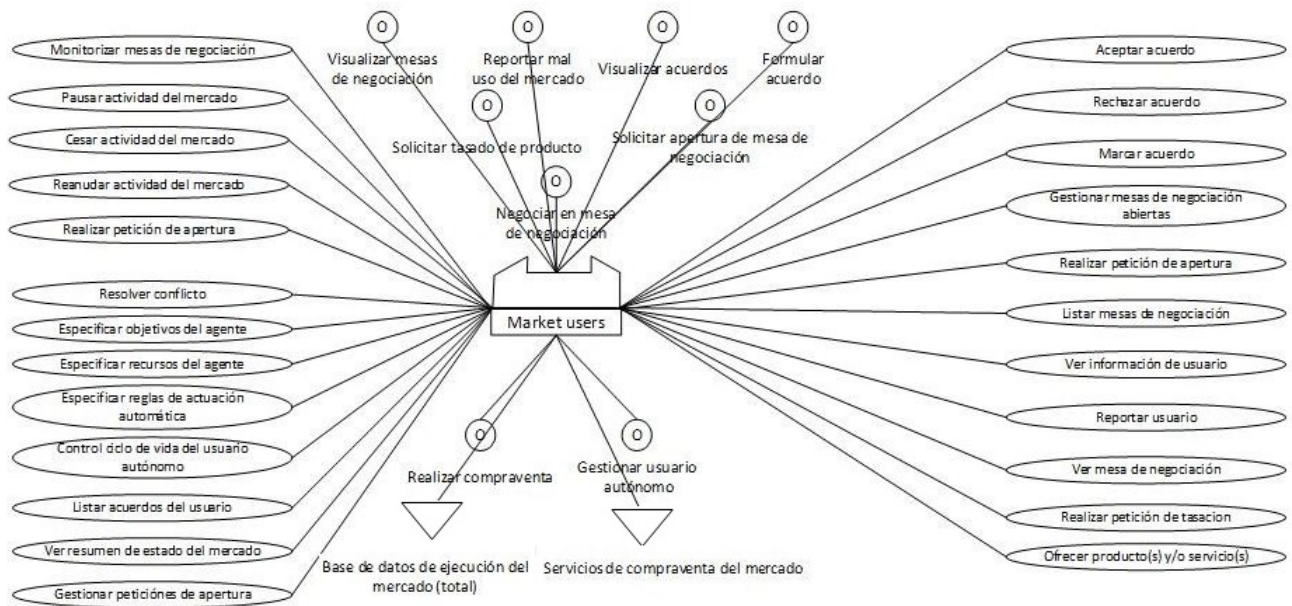


Figura 3.16: Diagrama que resume la funcionalidad del usuario del mercado

representativo de los usuarios que realizan la compraventa en el mercado se extienden en los roles representativos del comprador (*Buyer*) y vendedor (*Seller*).

### 3.2.2.3 Diagrama de interacción

Tras identificar los roles y su relación con las tareas anteriormente definidas se procede a visualizar la relación entre los distintos roles para realizar tareas. El objetivo es localizar aquellas tareas que requieran definir un protocolo posteriormente en la fase 4 de la metodología (en donde se modelizarán diagramas de interacción). El resultado de dicha visualización se puede encontrar en la Figura 3.18.

Podemos observar como muchas tareas solo son tratadas mediante un único rol, tales como la identificación, la gestión de diversas listas de la plataforma y la definición de objetivos del agente representante del usuario. Otras, requieren de la interacción entre dos roles, tales como el registro de usuarios. En otros casos, tal como la tarea *Monitorizar mesas de negociación*, la realización de la misma resulta un proceso complejo que requiere la interacción entre muchos roles (en este ejemplo, se requiere la interacción entre una o varias entidades con el rol *Market User*, una entidad con el rol *Table Judge* encargada de gestionar la mesa y una entidad con el rol *Market Manager* que gestione la ejecución de la mesa de negociación).

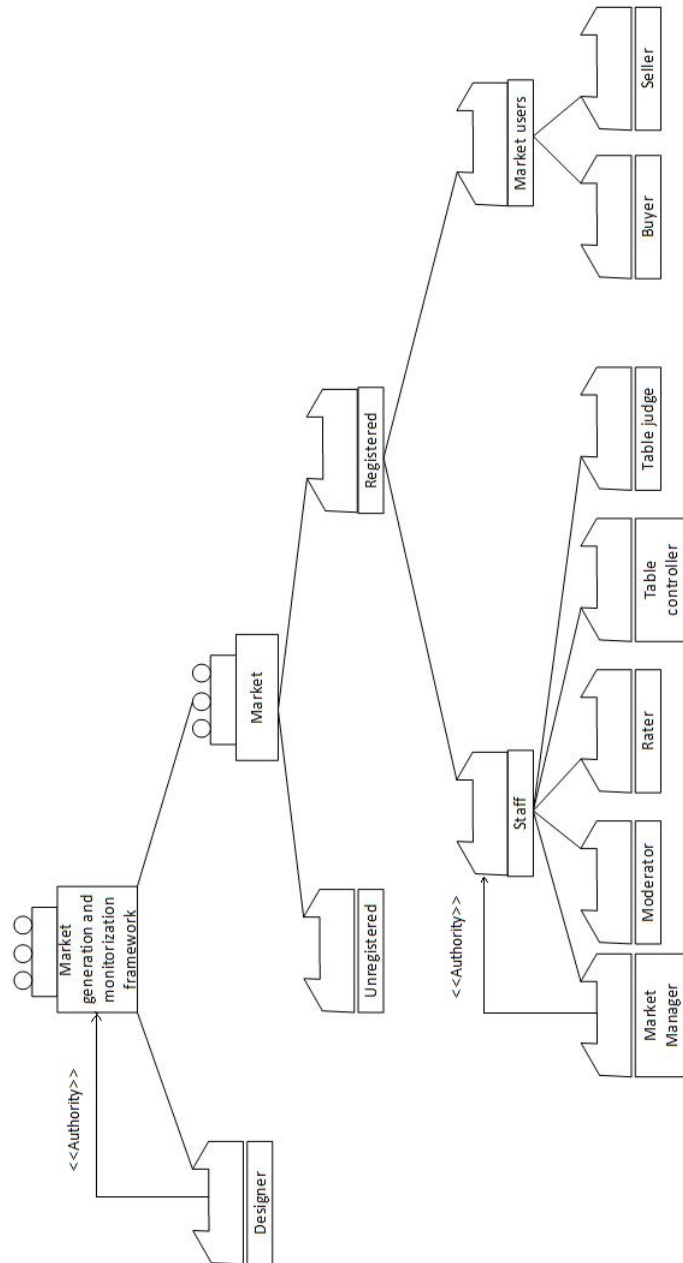


Figura 3.17: Estructura social y organizacional

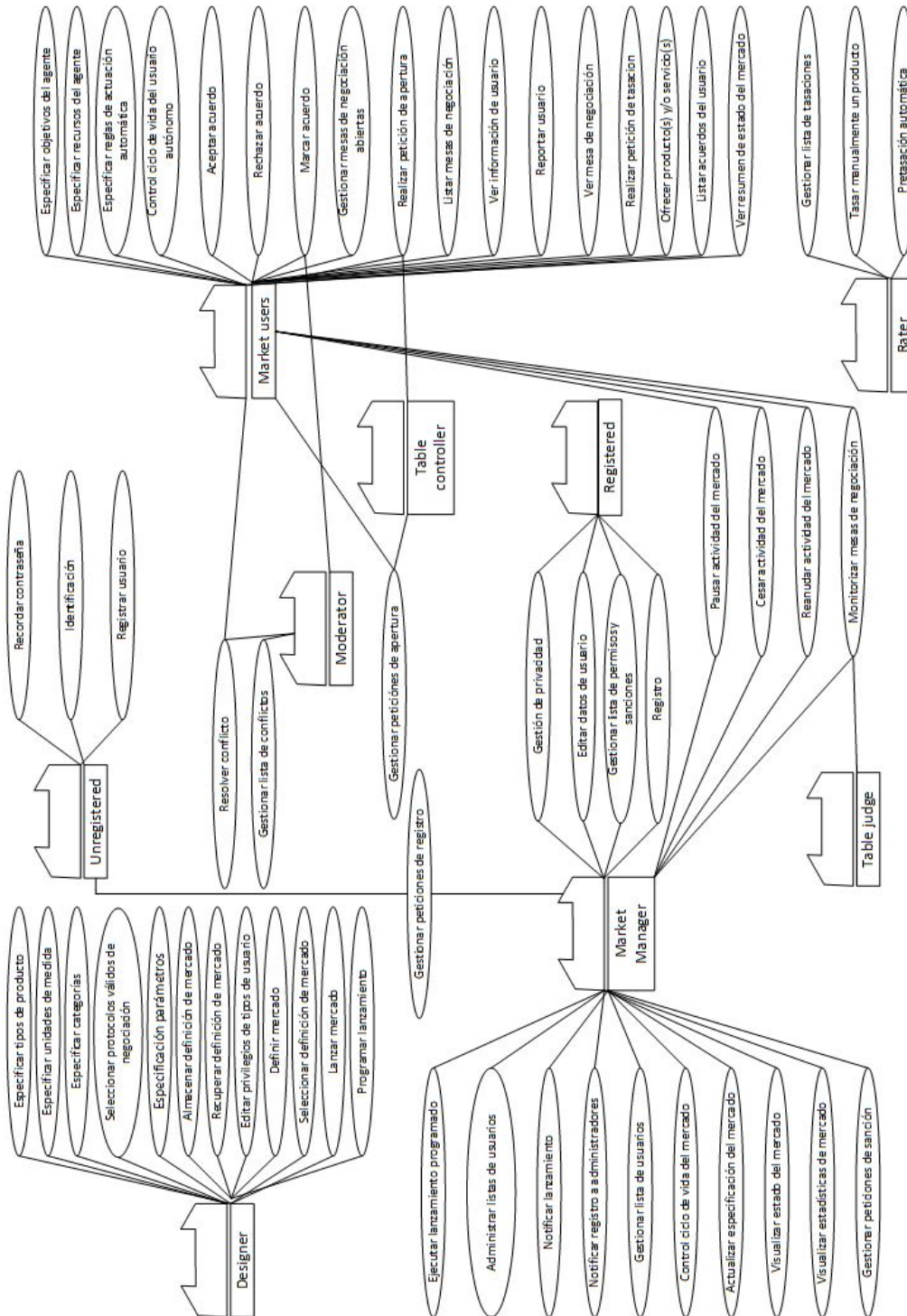


Figura 3.18: Diagrama que indica la interacción entre los distintos roles para la realización de tareas

En este punto finaliza la fase 2 de la metodología, en la que se ha definido la organización y los roles que lo representan y se dispone a especificar el contexto normativo que empleará el sistema a producir (fase 3).

### **Fase 3: Especificación del contexto normativo**

Tras la definición desde un alto nivel de la organización, su estructuración en tareas y definición de la organización que gestionará el sistema se especifica el contexto normativo en el que se englobará la organización y que facilitará la formalización de los acuerdos alcanzados en la plataforma mediante la creación de contratos. Como consecuencia de la realización de esta fase, se han producido los siguientes documentos:

1. *Guía de normas de la organización*: guía que identifica las restricciones en el comportamiento de las entidades que forman el sistema en base de la especificación del sistema en las fases anteriores (Epígrafe 3.2.2.1).

#### **3.2.2.1 Guía de normas de la organización**

Siguiendo la guía para la identificación de normas de ROMAS, se han recogido las posibles restricciones al comportamiento de las entidades del sistema dentro de la plataforma. Estas normas se definen de forma sencilla mediante cinco propiedades que se explican a continuación:

- *Deontic*, o acción de la norma: indica qué acción es la que hace la norma (PERMITIR, PROHIBIR u OBLIGAR).
- *Target*, o roles implicados: indica qué roles están implicados en esta norma y les afecta.
- *Action*, o acción a realizar: indica qué acción o acciones se desencadenan a partir de la activación de la norma.
- *Activation*, o suceso activador: indica qué sucesos provocan la activación de la norma.
- *Expiration*, o suceso desactivador: indica qué sucesos provocan la desactivación de la norma.

El Cuadro 3.2 muestra algunas de las normas que se han detectado empleando la guía de la metodología.

Deontic	Target	Action	Activation	Expiration
PERMITIR	<i>Registered</i>	Acceso al mercado	El usuario inicia la sesión	Fin de sesión
PROHIBIR	<i>Unregistered</i>	Acceso al mercado	Petición sin identificarse	-
PERMITIR	<i>Market manager</i>	Administrar plataforma	Iniciar sesión como administrador	Finalizar sesión
PROHIBIR	<i>Market user</i>	Administrar plataforma	Iniciar sesión como administrador	-
PERMITIR	<i>Market manager</i>	Acceder a base de datos	Petición de acceso a BD	Fin de acceso
PROHIBIR	<i>Unregistered</i>	Acceder a base de datos	Petición de acceso a BD	-

Cuadro 3.2: Ejemplo de normas generadas para la plataforma

#### Fase 4: Especificación de las actividades

Tras la especificación de los roles y la organización para resolver las tareas, identificar las interacciones entre los distintos roles en tareas específicas y la gestión de la normativa del sistema se procede ahora a especificar con más detalle las tareas que se realizarán en el sistema, los protocolos de interacción empleados en las tareas en las que se relacionan dos o más roles y los servicios que ofrecerá el sistema. Como resultado de esta fase se ha obtenido lo siguiente

1. *Diagramas de interacción*: que identifican el protocolo a seguir por más de un rol para realizar tareas concretas (Epígrafe 3.2.2.1).

##### 3.2.2.1 Diagramas de interacción

En la Fase 2 se definieron los roles que realizarán las tareas que definen el sistema. Gracias al diagrama de interacción pudimos visualizar las tareas y/o subtareas en las que dos o más roles actuaban al mismo tiempo y requieren un protocolo para realizarlas correctamente y sin conflictos. Por otra parte, el control normativo dado en la fase 3 permite conocer restricciones que se aplicarán al conjunto del sistema. De estas restricciones normativas algunas afectan a los protocolos definidos. En esta producción de la fase se procede a detallar los protocolos para reflejar el contexto normativo dado y especificar las interacciones del sistema. Para ello se emplearán diagramas de interacción. Los diagramas de interacción se asemejan a los equivalentes a los definidos en el lenguaje de modelado UML [18]. Su principal diferencia es que la transmisión de mensajes se realiza entre roles y no entre actores u objetos. A

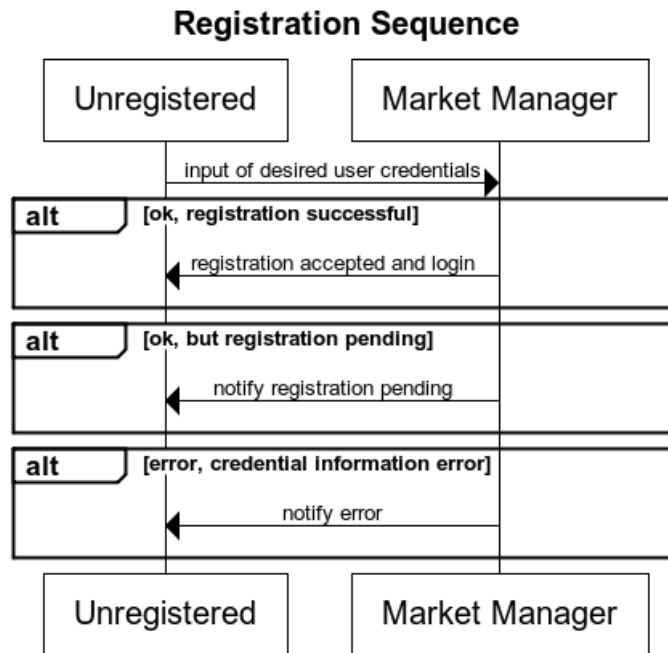


Figura 3.19: Diagrama de interacción correspondiente al registro de un usuario

continuación se detallan estos diagramas, estructurados en función de la parte de la plataforma de mercado que representan.

### 3.2.2.1.1 Acreditación de usuarios

En primer lugar, los protocolos que se han identificado en el proceso de acreditación de usuarios dentro de la plataforma se pueden observar en los diagramas de interacción. En la Figura 3.19 podemos observar el proceso de registro de un usuario de una plataforma en el que un usuario envía los datos de registro al gestor del mercado y el gestor del mercado responde al usuario no registrado en función de si es posible el registro o no. En la Figura 3.20 podemos observar el proceso de identificación de usuarios dentro de la plataforma dadas las credenciales obtenidas en el registro. Dependiendo de si el usuario dispone de las credenciales correctas se le informa si la identificación es correcta o no. En la Figura 3.21 se detalla el diagrama de interacción que trata el protocolo a seguir tras la identificación del usuario, en el que el gestor del mercado informa al usuario registrado de su rol dentro de la plataforma.

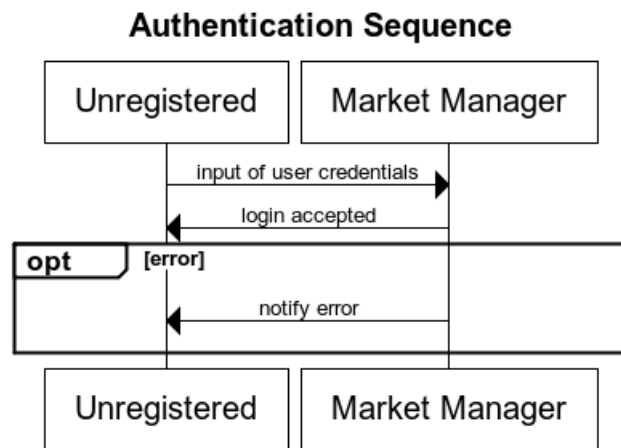


Figura 3.20: Diagrama de interacción correspondiente a la identificación de usuarios



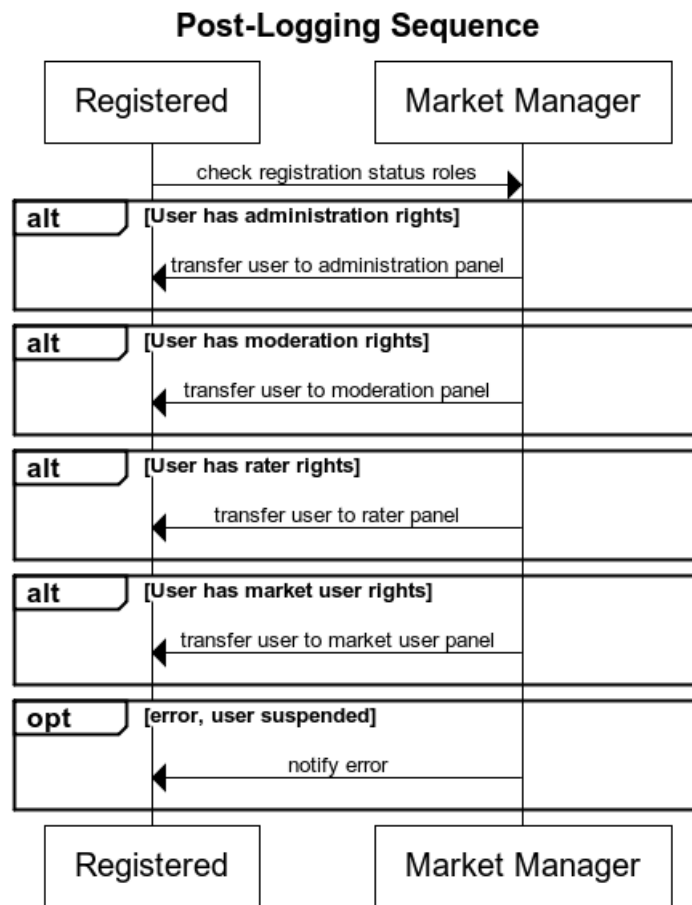


Figura 3.21: Diagrama de interacción correspondiente al proceso después de la identificación de un usuarioki

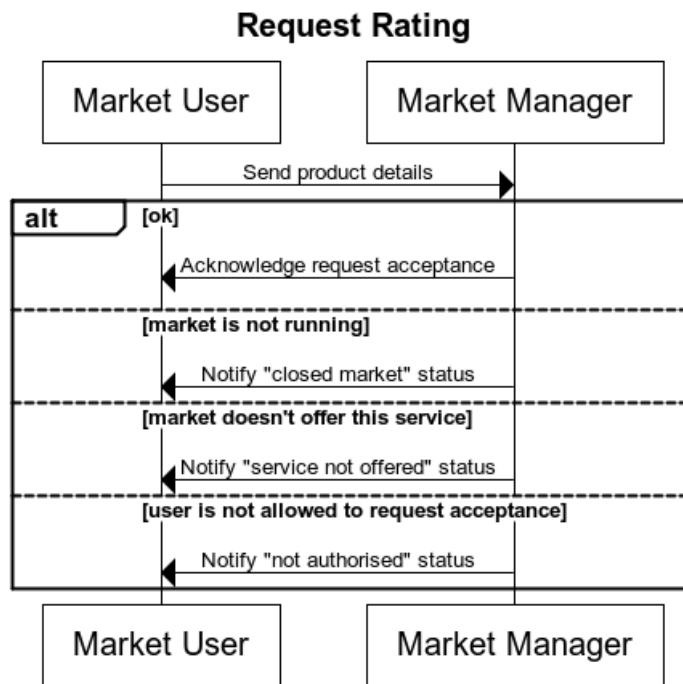


Figura 3.22: Diagrama de interacción correspondiente a la solicitud de una tasación

### 3.2.2.1.2 Tasación de productos y servicios

En la funcionalidad correspondiente a la tasación, el Diagrama 3.22 se observa el proceso que realiza un usuario de mercado para pedir una tasación y recibir la confirmación de su recepción. En el diagrama correspondiente a la Figura 3.23 se representa la solicitud de una lista de peticiones de tasación al manejador de mercado. En la Figura 3.24 se visualiza la solicitud de los detalles de una petición de tasación en concreto al gestor de mercado. Por último en la Figura 3.25 se resuelve la petición de tasación de un producto y envía al *Market Manager* los datos de la resolución.

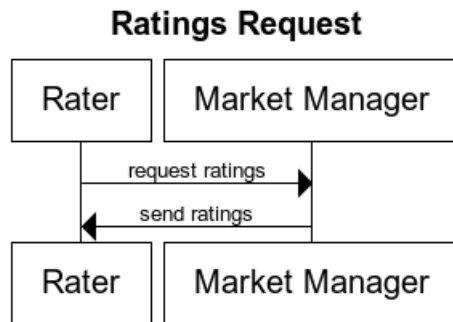


Figura 3.23: Diagrama de interacción correspondiente a la petición del listado de peticiones de tasado

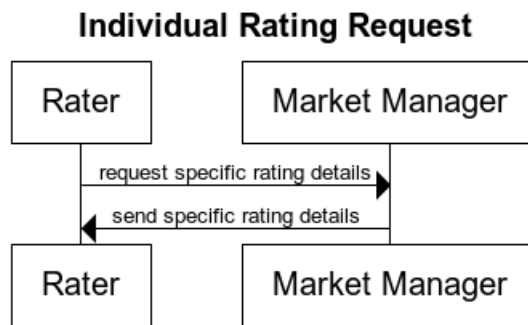


Figura 3.24: Diagrama de interacción correspondiente a la solicitud de los detalles de una petición de tasación

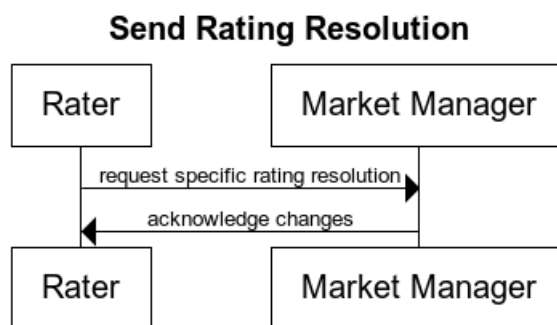


Figura 3.25: Diagrama de interacción correspondiente a la resolución de una petición de tasación

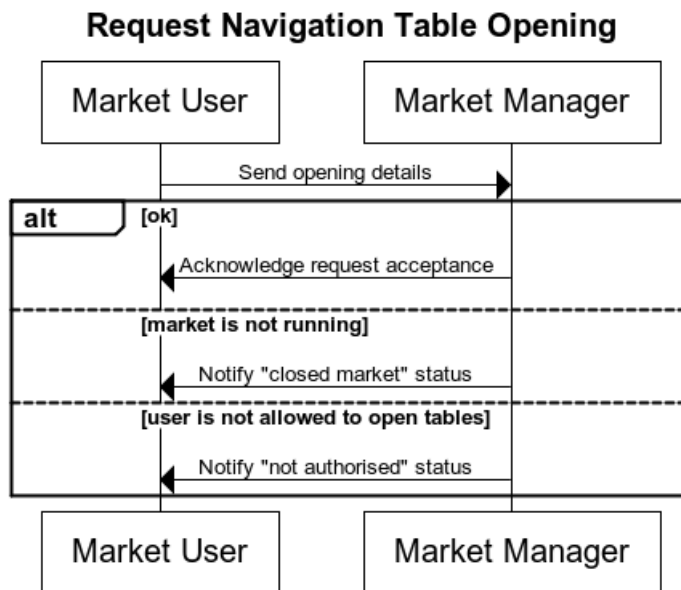


Figura 3.26: Diagrama de interacción correspondiente a la petición de la apertura de una mesa de negociación

### 3.2.2.1.3 Apertura de mesas de negociación

En cuanto a la apertura de mesas de negociación, el proceso de petición de apertura de una mesa de negociación a un agente con el rol *Table Controller* se puede observar en la Figura 3.26, su resolución e informe de la misma en la Figura 3.28 y el informe de un cambio en como responder ante una solicitud de una mesa de negociación en la Figura 3.29 (permitiendo así la apertura automática de las mesas o denegar automáticamente la apertura de estas).

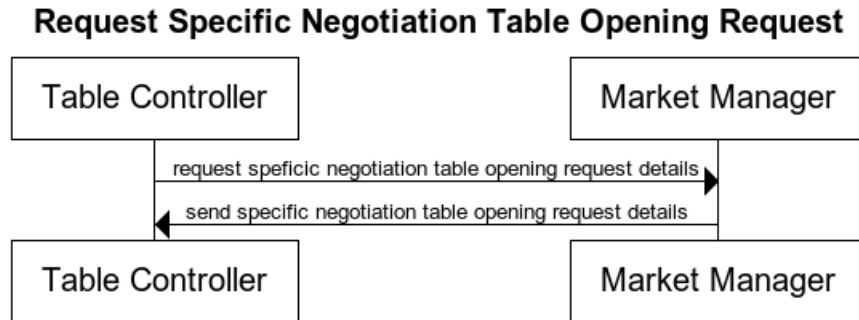


Figura 3.27: Diagrama de interacción correspondiente a la solicitud de los detalles de una petición de apertura de una mesa de negociación

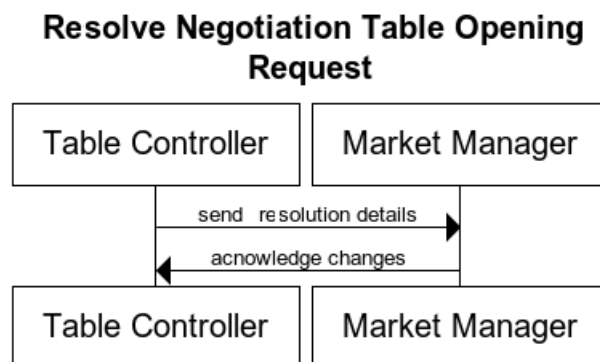


Figura 3.28: Diagrama de interacción correspondiente a la notificación de la resolución de una petición de apertura de una mesa de negociación

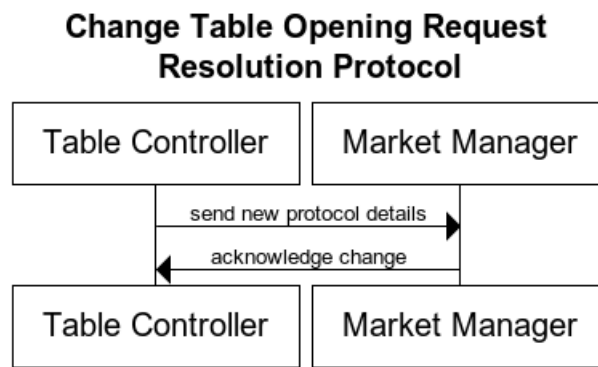


Figura 3.29: Diagrama de interacción correspondiente a la notificación de cambio del protocolo de resolución de peticiones de apertura de mesas de negociación



Figura 3.30: Diagrama de interacción correspondiente a la petición de la lista de mesas de negociación abiertas

#### 3.2.2.1.4 Proceso de negociación dentro de mesas de negociación

En cuanto a los diagramas de interacción correspondientes a la negociación en las mesas de negociación, en la Figura 3.30 se visualiza la petición de un listado de las mesas de negociación abiertas a un gestor del mercado. En la Figura 3.31 se observa la petición de la información correspondiente a una mesa de negociación en detalle. En la Figura 3.32 se puede apreciar el marco en el cual se realizarán las negociaciones de forma genérica sin contar la parte dependiente de protocolo. Este protocolo podría ser, por ejemplo, el especificado en la Figura 3.33 (se trata de una subasta japonesa basada en el protocolo *FIPA ACL Japanese Auction*). Por último y tras la formalización de acuerdos, la Figura 3.34 permite observar la interacción entre un usuario y el gestor de mercado para el cambio del estado del acuerdo.

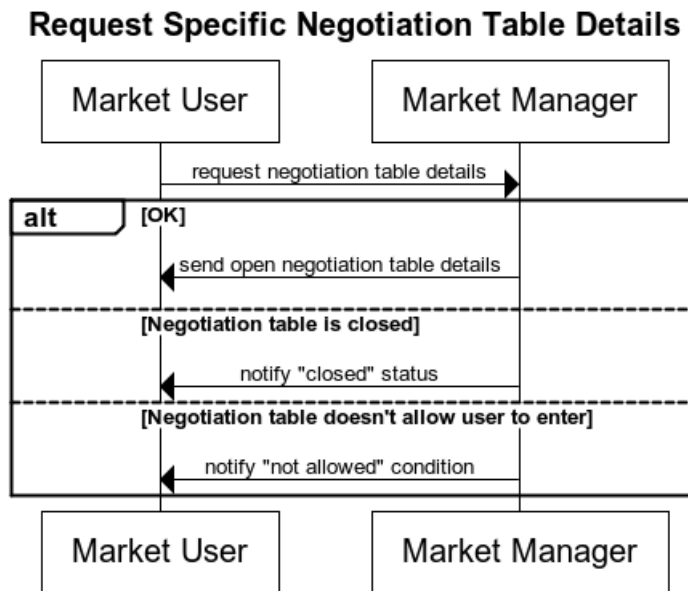


Figura 3.31: Diagrama de interacción correspondiente a la solicitud de los detalles de una mesa de negociación específica



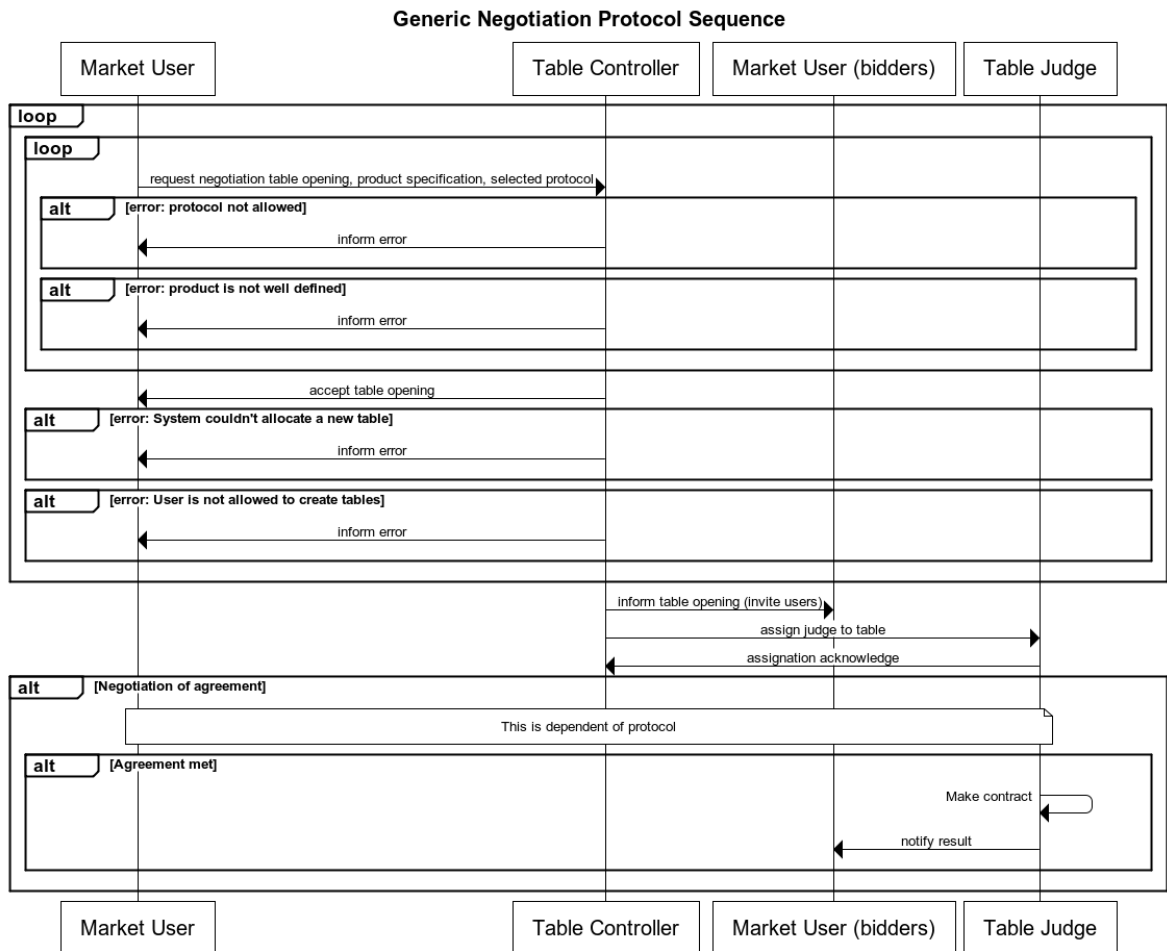


Figura 3.32: Diagrama de interacción correspondiente a la negociación genérica de acuerdos

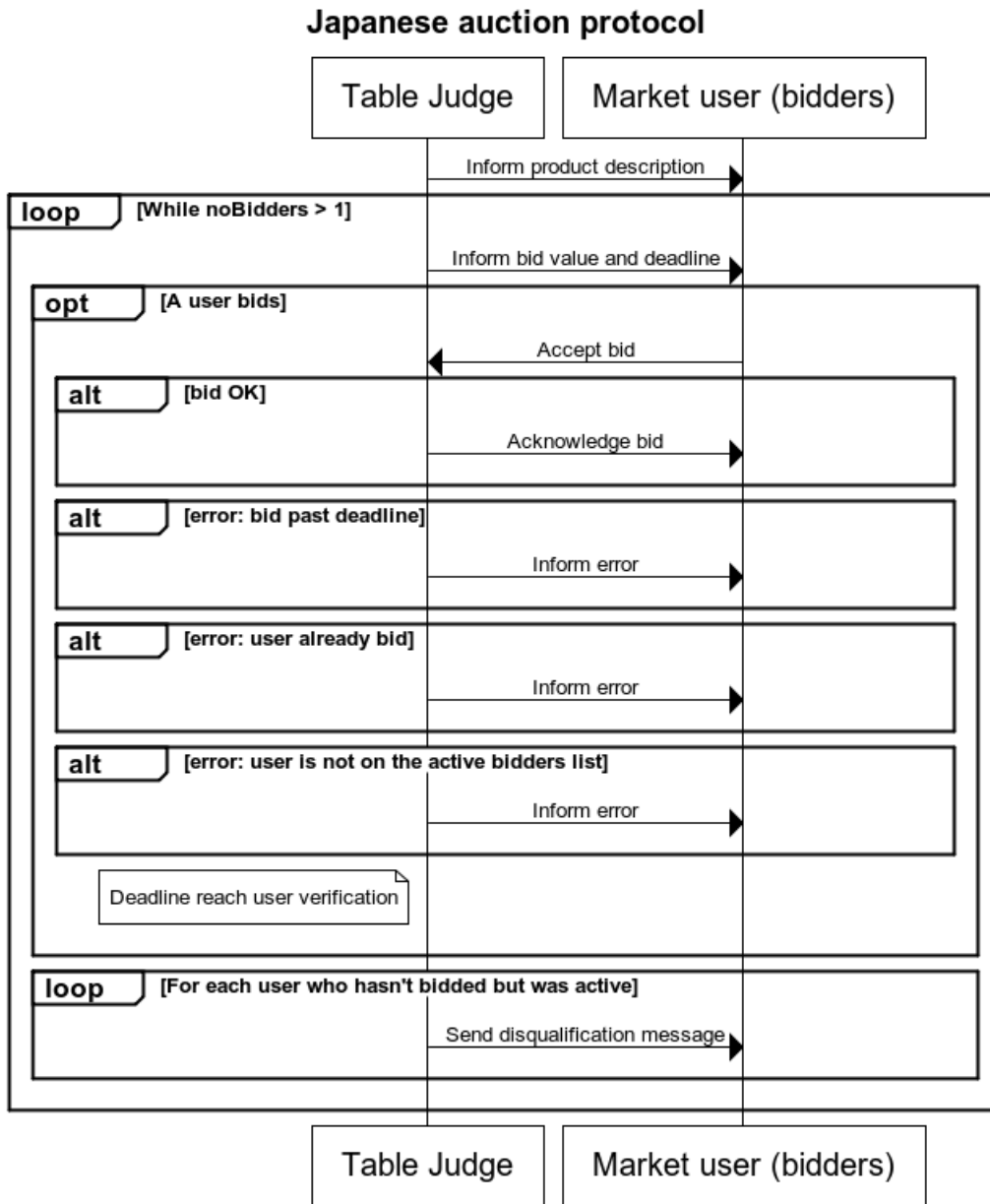


Figura 3.33: Diagrama de interacción correspondiente al protocolo de negociación de subasta japonesa

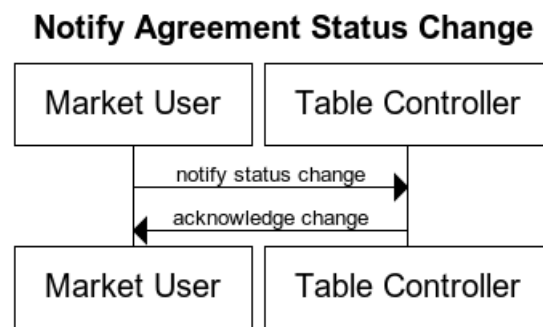


Figura 3.34: Diagrama de interacción correspondiente a la notificación del cambio en el estado del acuerdo

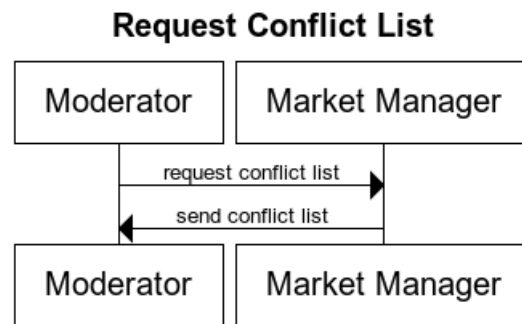


Figura 3.35: Diagrama de interacción correspondiente a la petición de la lista de conflictos



Figura 3.36: Diagrama de interacción correspondiente a la solicitud de los detalles de un conflicto específico

#### 3.2.2.1.5 Resolución de conflictos

En cuanto a la resolución de conflictos, la Figura 3.35 indica la interacción necesaria entre un moderador y el gestor del mercado para la obtención de una lista de conflictos, la Figura 3.36 detalla la petición de los detalles de un conflicto específico al gestor del mercado y la Figura 3.37 muestra la notificación de la resolución de un conflicto enviando un moderador la información de la resolución al gestor del mercado.



Figura 3.37: Diagrama de interacción correspondiente a la notificación de la resolución de un conflicto

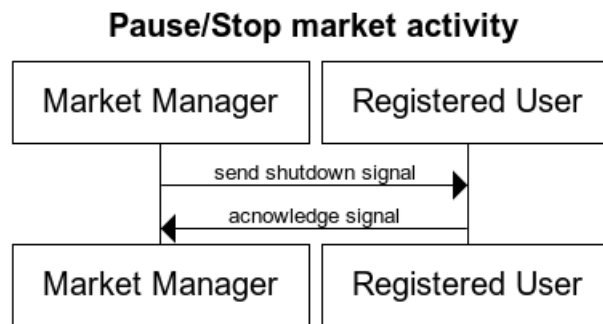


Figura 3.38: Diagrama de interacción correspondiente a la parada de agentes debido a la pausa/parada de la actividad de la plataforma de mercado

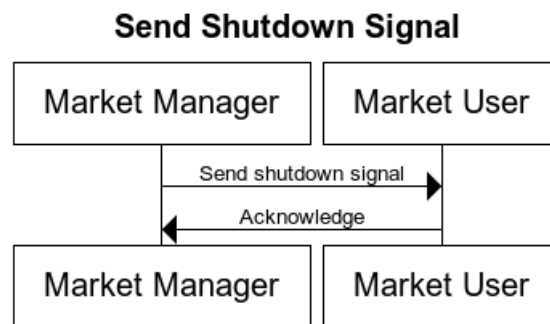


Figura 3.39: Diagrama de interacción correspondiente al envío de una señal de desactivación de agente

#### 3.2.2.1.6 Ciclo de vida del mercado

Por último se detallan las interacciones correspondientes a la parada del mercado por parte del gestor de mercado (tanto su parada como su desactivación). Las Figuras 3.38 y 3.39 detallan el proceso en el cual envía un gestor del mercado una señal uno por uno a todos los usuarios del mercado para indicar la finalización temporal o permanente de la actividad en el mercado.

### Fase 5: Especificación de los agentes

La fase 5 de la metodología aplica todo el conocimiento obtenido en las anteriores fases para diseñar los agentes que formarán el sistema multiagente que representará el

sistema. Este diseño y su posterior implementación se tratarán en el capítulo orientado a la implementación del sistema multiagente (Capítulo 4).

En este punto finaliza el proceso de análisis y diseño de la plataforma, en la que se ha especificado como debe ser el sistema que se propone implementar empleando un modelo de negocio y la metodología ROMAS. A continuación se detalla la implementación del sistema.

# Capítulo 4

## Implementación

En este capítulo se introducen las herramientas escogidas para el desarrollo de un prototipo que muestre las funcionalidades del entorno diseñado y el proceso de desarrollo del mismo. Se concluirá con la presentación de los resultados obtenidos de la implementación.

### 4.1. Introducción

Tras el análisis y diseño detallados en los capítulos anteriores prosigue el desarrollo del trabajo con la implementación de un prototipo que ilustre el diseño realizado y permita observar una interacción con el usuario empleando sistemas multiagente y las tecnologías del acuerdo detalladas en la introducción. Para ello se ha diseñado una arquitectura similar a las dirigidas a aplicaciones comerciales orientadas a mercados virtuales en las que se permite la interacción por parte del usuario por distintos dispositivos y plataformas además de asegurar su persistencia mediante el uso de una o varias bases de datos. Así, tal como se observa en la Figura 4.1, la arquitectura emplea como base el entorno de desarrollo facilitado por MAGENTIX2 (explicado con más detalle en la Sección 4.3) para dar soporte al sistema multiagente y a la organización que gestiona el mercado. Para mantener la persistencia de datos y dado que es una dependencia de MAGENTIX2 se emplea una base de datos relacional en lenguaje SQL y con libertad de elección de plataforma (en las que se incluyen mySQL, Oracle DBMs o Microsoft SQL Server, entre otras).



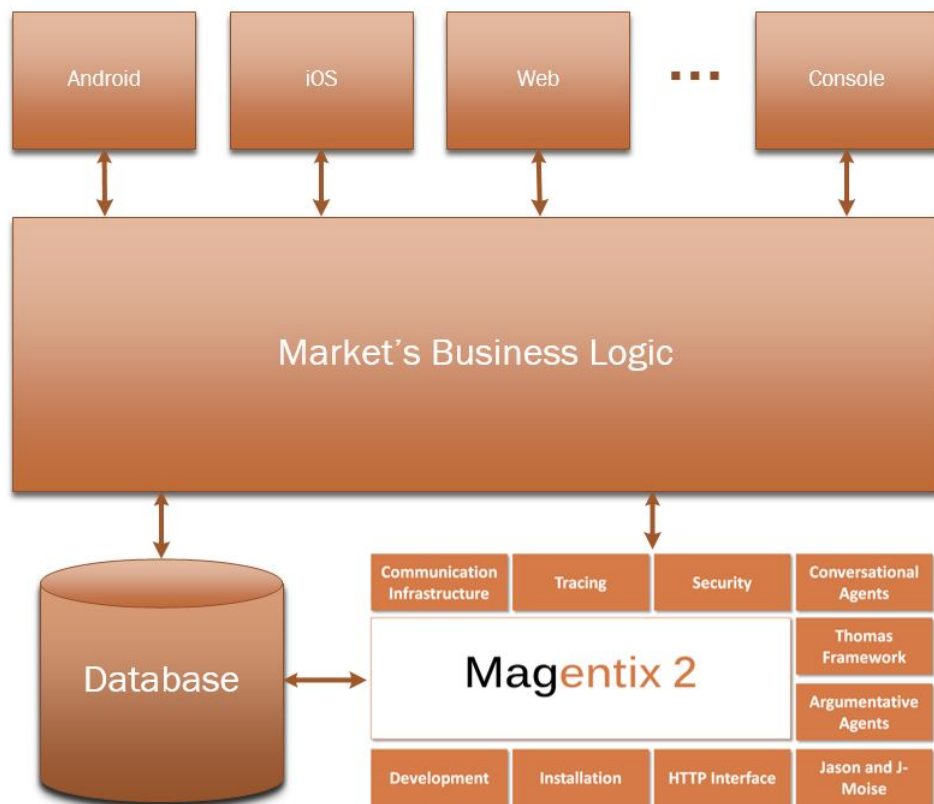


Figura 4.1: Arquitectura del mercado genérico virtual implementado

La interacción entre la capa de persistencia y el usuario se completa mediante la lógica de negocio de la aplicación y la interfaz gráfica al usuario. La lógica de negocio versa de un sistema multiagente en la cual las entidades del sistema (los agentes) interactúan con el mercado para cumplir sus objetivos individuales empleando las funcionalidades ofrecidas por el entorno de desarrollo. Por último la interacción entre el usuario y la lógica de negocio se realiza mediante el uso de una interfaz del usuario. Dependiendo de las necesidades del usuario y dada la libertad que deja la lógica de negocio se puede desarrollar una interfaz de usuario mediante una página web, mediante aplicaciones específicas para sistemas operativos móviles, una aplicación de escritorio, el uso de una terminal, etcétera.

A continuación se detallan las herramientas empleadas durante la implementación del prototipo.

## 4.2. Java

El lenguaje de programación Java es un lenguaje de naturaleza imperativa cuyas ventajas son la facilidad en la portabilidad del código producido a distintas plataformas, su extenso catálogo de librerías disponibles para su uso de forma libre y gratuita durante el desarrollo además de su ejecución en una máquina virtual aislada de la ejecución de otros programas (*sandbox*). Como desventajas se destaca su sobrecoste computacional debido a la máquina virtual anteriormente citada, su complejidad para el desarrollo en diversos aspectos como el paradigma de reflexión de código y la falta de diversos mecanismos comunes en los lenguajes de programación imperativos como el uso de punteros.

Se ha escogido el lenguaje Java principalmente por la adopción que éste ha tenido en la industria, la disponibilidad de librerías externas que facilitan el desarrollo del trabajo y por su portabilidad a nivel de sistema operativo.

## 4.3. MAGENTIX2

Para el desarrollo de plataformas de agentes y ofrecer servicios relacionados con los mismos, se emplean frameworks de desarrollo orientados a agentes. Ejemplos de

estos frameworks de desarrollo son JADE [4], MadKit [10] y Andromeda [1], entre otros [17]. MAGENTIX2 [2] es un framework desarrollado por el Grupo de Tecnología Informática – Inteligencia Artificial en la Universidad Politécnica de Valencia orientado a ofrecer soporte para la implementación de sistemas multiagente abiertos y distribuidos. Su principal característica es su portabilidad a distintos tipos de sistema dado que ha sido desarrollado en el lenguaje de programación Java y no emplea librerías dependientes de un sistema operativo específico. También destaca por la amplia gama de funcionalidades que ofrece al programador de sistemas multiagente para la creación de soluciones informáticas.

En cuanto al soporte de la plataforma se ofrecen servicios a nivel de organización, ofreciendo tecnologías que permiten el control de sociedades de agentes; servicios a nivel de interacción, con facilidades orientadas a la comunicación entre agentes y servicios a nivel de agente, con facilidades orientadas al razonamiento lógico empleando arquitecturas de razonamiento basadas en el modelo BDI y al aprendizaje automático. En este trabajo se aplica el uso de las facilidades a nivel de organización con el objetivo de crear la organización que gestionará el funcionamiento del mercado creado, a nivel de interacción para definir los protocolos de negociación en las mesas de negociación y en las comunicaciones entre agentes para poder realizar acciones en el mercado y, finalmente, se emplea a nivel de agentes para fijar los objetivos personales de los agentes automáticos y el razonamiento de los mismos en base a un conjunto de planes precompilados.

En cuanto a funcionalidades la plataforma emplea el protocolo de intercambio de mensajes AMQP [24]. El protocolo permite la interacción entre agentes dentro de la plataforma de forma eficiente y transparente para el programador. Su soporte de organizaciones es otro de los factores clave para su elección en la implementación del mercado. Se puede definir una organización a nivel estructural definiendo roles dentro de la organización, asignando roles a los distintos roles del sistema y finalmente emplear un servicio de páginas blancas para identificar agentes con roles específicos dentro de la organización.

Por otra parte cabe destacar su soporte para agentes conversacionales y el desarrollo de factorías conversacionales, que permiten paralelizar la interacción entre los agentes que forman el sistema mediante el uso de hilos además de facilitar la imple-



Figura 4.2: Logotipo y esquema ilustrativo de las funcionalidades de la plataforma MAGENTIX2

mentación de protocolos de negociación definiendo plantillas (factorías) que definan el flujo de acciones del agente en base a un autómata finito que toma como base las interacciones realizadas sobre el entorno y las respuestas recibidas. Otras funcionalidades de la plataforma que han condicionado su elección orientadas al trabajo futuro son su soporte a agentes que emplean Jason que permiten razonar empleando un razonamiento basado en arquitecturas BDI, el uso de agentes argumentativos y adaptadores HTTP que permiten facilitar la integración de los sistemas multiagente desarrollados en la plataforma con aplicaciones web o aplicaciones móviles (empleando *web services*), entre otros.

Se puede obtener más información sobre este framework de desarrollo y proyectos construidos a partir de este entorno en [2].

## 4.4. Otras herramientas

Por completitud y con el objetivo de ilustrar las tecnologías empleadas para la implementación del prototipo se detallan otras herramientas empleadas para esta fase del trabajo. Entre estas herramientas destacan:

- **IntelliJ IDEA** (<http://www.jetbrains.com/idea>): Entorno de desarrollo creado por JetBrains Inc. (<http://www.jetbrains.com>) que permite facilitar y acelerar el

desarrollo de los componentes necesarios para implementar el prototipo. Ofrece funcionalidades similares a otros entornos de desarrollo como Eclipse o NetBeans además de otras útiles para este trabajo como el soporte de frameworks orientados a la gestión de bases de datos, inspección de código y soporte nativo de sistemas de control de versiones.

- **Hibernate** (<http://www.hibernate.org/>): Framework que permite gestionar con facilidad la interacción entre la capa lógica del sistema con la capa de persistencia (base de datos relacional). Permite convertir un esquema de base de datos en clases Java que los representa, además de facilitar la conexión con las distintas bases de datos.
- **MySQL** (<http://www.mysql.com/>): Sistema gestor de bases de datos desarrollado por Oracle. Su uso ha permitido el desarrollo de la capa de persistencia de datos del sistema implementado mediante el uso del lenguaje de definición de modelos relacionales SQL.
- **SQLWorkbench** (<http://www.mysql.com/products/workbench/>): Herramienta creada por Oracle que permite el diseño y gestión de modelos de datos entidad-relación y el control de bases de datos MySQL.
- **PHPMyAdmin** (<http://www.phpmyadmin.net/>): Herramienta software que permite gestionar bases de datos MySQL de forma sencilla a través de un navegador.
- **Subversion** (<https://subversion.apache.com/>): Sistema de control de versiones que permite disponer de un historial de versiones del sistema implementado, permitiendo el trabajo conjunto entre varios desarrolladores.
- **Redmine** (<http://www.redmine.org/>): Sistema de gestión de tareas que permite el control del progreso de la implementación en base a tareas (o *tickets*). Su uso, además, permite el control y gestión de los errores detectados en la implementación y su posterior resolución.

## 4.5. Del diseño a la implementación

En la transición de la etapa de diseño a la implementación del prototipo de plataforma de mercado se consideró la idea de emplear herramientas comunes en el desarrollo de software para complementar las producciones generadas mediante la metodología ROMAS. Para ello se han realizado dos tareas. En primer lugar, se consideró emplear un diagrama entidad-relación representado a partir de un diagrama de clases UML para diseñar un modelo de datos que permita gestionar la capa lógica del mercado y su almacenamiento en la capa de persistencia de la aplicación empleando una base de datos relacional. El resultado se muestra en la Sección 4.5.1. Seguidamente y para completar la validación del modelo generado en la metodología ROMAS y su adecuación a los requisitos propuestos para el trabajo se ha diseñado la interacción entre el usuario y la máquina mediante interfaces, las cuales se muestran a continuación en la Sección 4.5.2 en forma de *mockups*.

### 4.5.1. Diagrama de clases

Con el objetivo de estructurar, estandarizar y jerarquizar la información tratada dentro de la plataforma de mercado se diseñó un diagrama entidad-relación que representa los datos tratados dentro de la misma. Dicho diagrama se puede observar en la Figura 4.3. El diagrama define como base el mercado junto a sus atributos (entidad *Market*) y sus propiedades a nivel estadístico (entidad *MarketStatistic*). De esta definición se extienden los usuarios que forman parte de la plataforma (entidad *User*) las mesas de negociación (entidades *NegotiationTable*, *Protocol* y como ejemplo *JapaneseAuction*) y la definición de los productos (entidades *Product*, *Property* y *MeasuringUnit*) que se comerciarán en la plataforma se extienden de esta base de mercado. Por otra parte se da soporte lógico a los agentes que representan a los usuarios (entidades *UserAgent* y *Objective*). Por otro lado se da soporte a los objetos intermedios creados por el uso de los servicios de la plataforma tales como la tasación (entidad *RatingRequest*) y la moderación (entidad *ModerationRequest*). Por último se da soporte a la actividad del mercado y sus usuarios ofreciendo un soporte lógico que registre las interacciones dentro de las mesas de negociación (entidad *ProtocolAction*) y los acuerdos alcanzados (entidad *Agreement*) así como la gestión de los acuerdos

alcanzados (entidad *MarketAgreementStatus*).

### 4.5.2. Diseño de la interfaz gráfica de usuario

Como elemento adicional a la metodología y con el objetivo de facilitar la implementación de la plataforma, comprobar que existe coherencia entre la especificación dada por los encargados del proyecto y las producciones generadas por la metodología se realizaron *mockups* definiendo la interfaz de usuario de la plataforma creada. Éstos *mockups* sirven como base para corroborar la coherencia entre el diseño realizado con la metodología y es versátil en cuanto al medio de interacción implementado. Esto incluye la implementación mediante la creación de una página web que permita la interacción con los usuarios, mediante una aplicación orientada a dispositivos móviles en plataformas como *Android* o *iOS*, mediante una aplicación de escritorio empleando frameworks como *qtJambi* o *jSwing* o por una simple terminal. Los *mockups* creados se categorizan en las distintos roles de la plataforma, a saber:

1. Panel de bienvenida y registro de usuarios no registrados
2. Administración del mercado
3. Administración de usuarios
4. Administración de tasaciones
5. Administración de conflictos
6. Administración de peticiones de apertura de mesas de negociación
7. Uso e interacción de la plataforma de mercado

#### Panel de bienvenida y registro

La primera página que el usuario del mercado podrá observar corresponde a la de la pantalla de bienvenida del mercado (Figura 4.4) en el cual el usuario podrá de forma muy similar a otras plataformas identificarse mediante un par usuario contraseña. Además, podrá acceder al panel de registro (Figura 4.5) para proceder a registrarse como usuario del mercado.

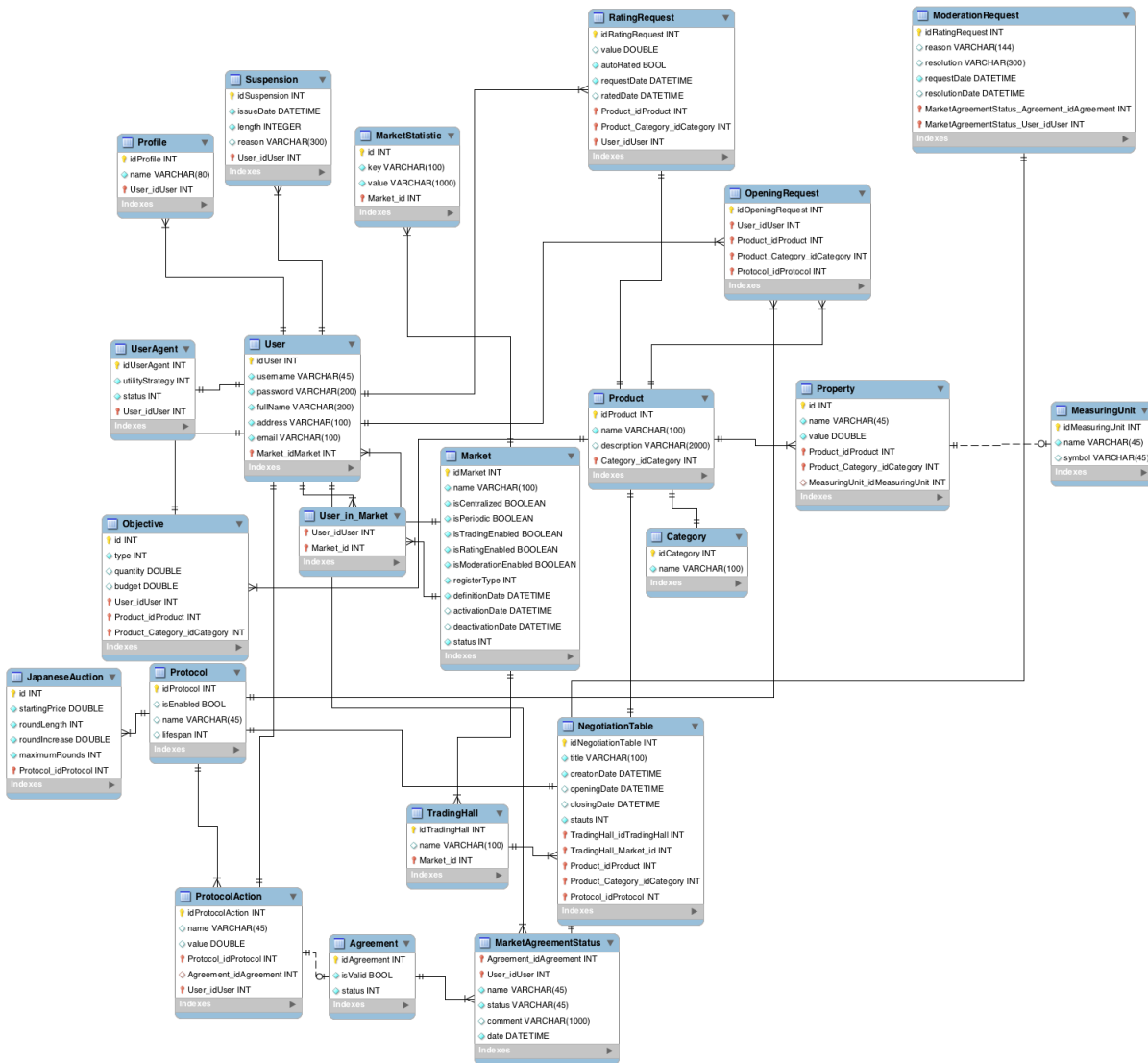


Figura 4.3: Diagrama de clases correspondiente a la plataforma de mercado



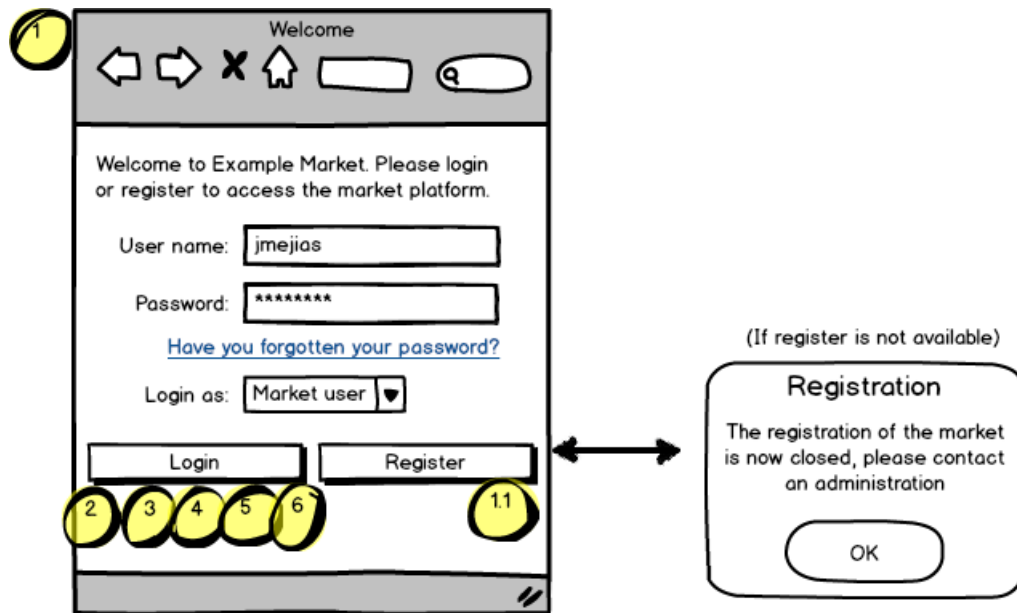


Figura 4.4: Prototipo de la interfaz correspondiente a la pantalla de bienvenida al usuario

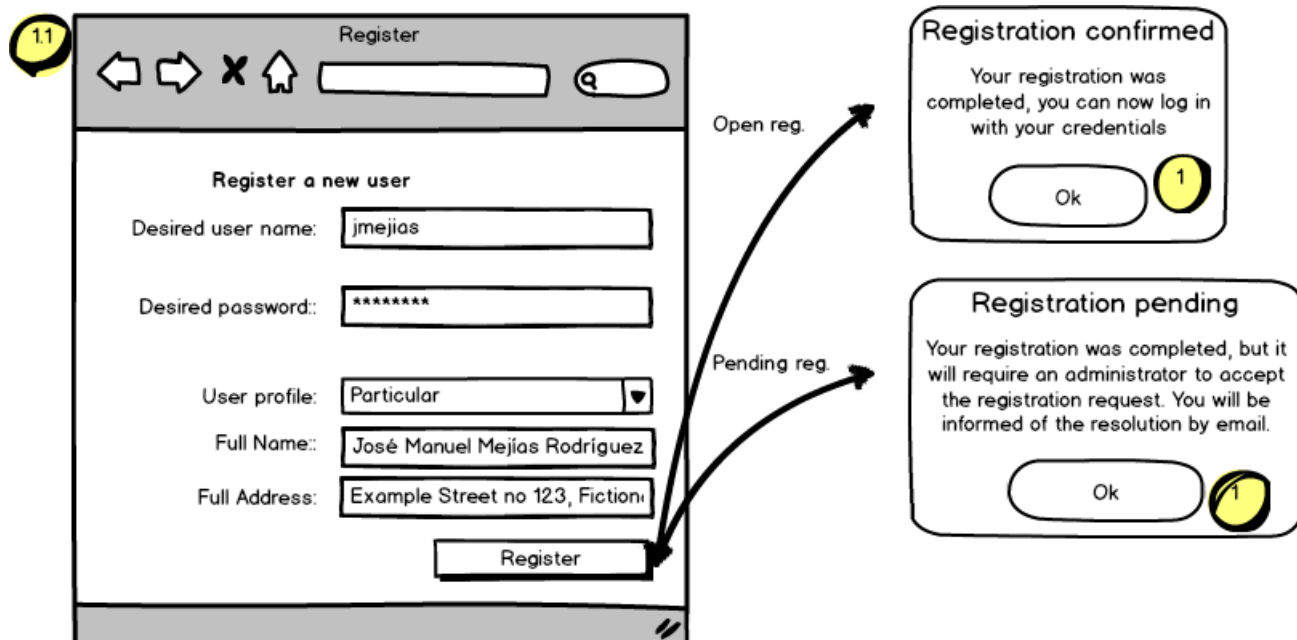


Figura 4.5: Prototipo de la interfaz correspondiente al registro del usuario

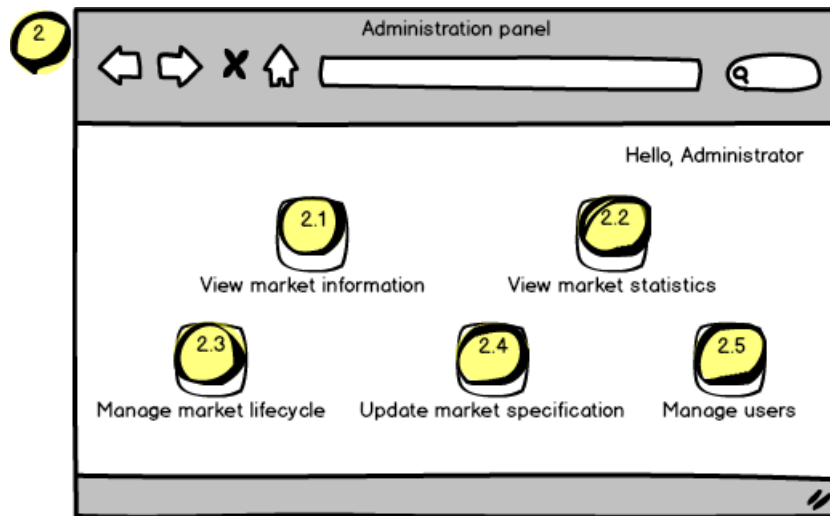


Figura 4.6: Prototipo de la interfaz correspondiente al panel de administración del gestor del mercado

### Panel de administración del mercado

Como administrador de mercado la primera página que verá es la del panel de usuario, que indica al usuario las posibles acciones que puede hacer dentro de la plataforma. Desde ahí, podrá acceder a la página correspondiente a la información del mercado (Figura 4.7) en la que el gestor podrá observar información general del estado del mercado en tiempo real. En la Figura 4.8 podemos observar como se pueden listar estadísticas sobre la ejecución del mercado a lo largo del tiempo. En la Figura 4.9 se puede visualizar el ciclo de vida del mercado y realizar las acciones para gestionar el mismo tales como pausar temporalmente el mercado, cesar de forma permanente su actividad o reanudar su ejecución si se ha pausado la actividad del mercado.

### Panel de administración de usuarios

La administración de usuarios de la plataforma de mercado empieza con un panel resumiendo las acciones disponibles para el usuario (Figura 4.10). Desde este panel se puede pasar a una página en la que se puede registrar usuarios de forma manual y similar a la que se realiza por parte del usuario. Se ha diseñado para poderse realizar tanto cuando los usuarios no pueden registrarse cuando el registro automático esté cerrado como para registrar usuarios con roles especiales (gestores de mercado,

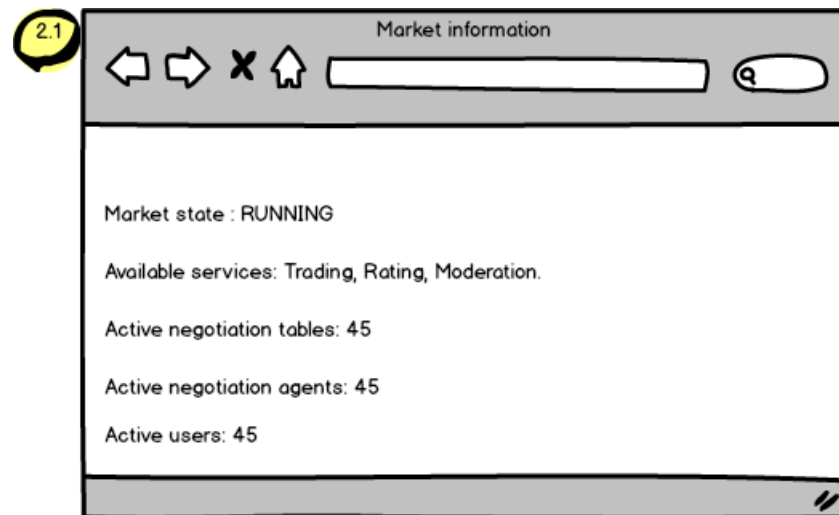


Figura 4.7: Prototipo de la interfaz correspondiente a la información del mercado

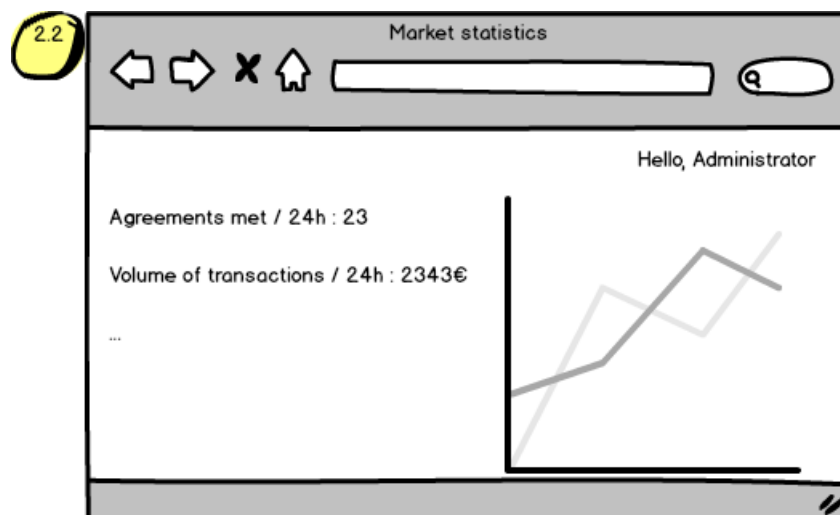


Figura 4.8: Prototipo de la interfaz correspondiente a la visualización de estadísticas del mercado

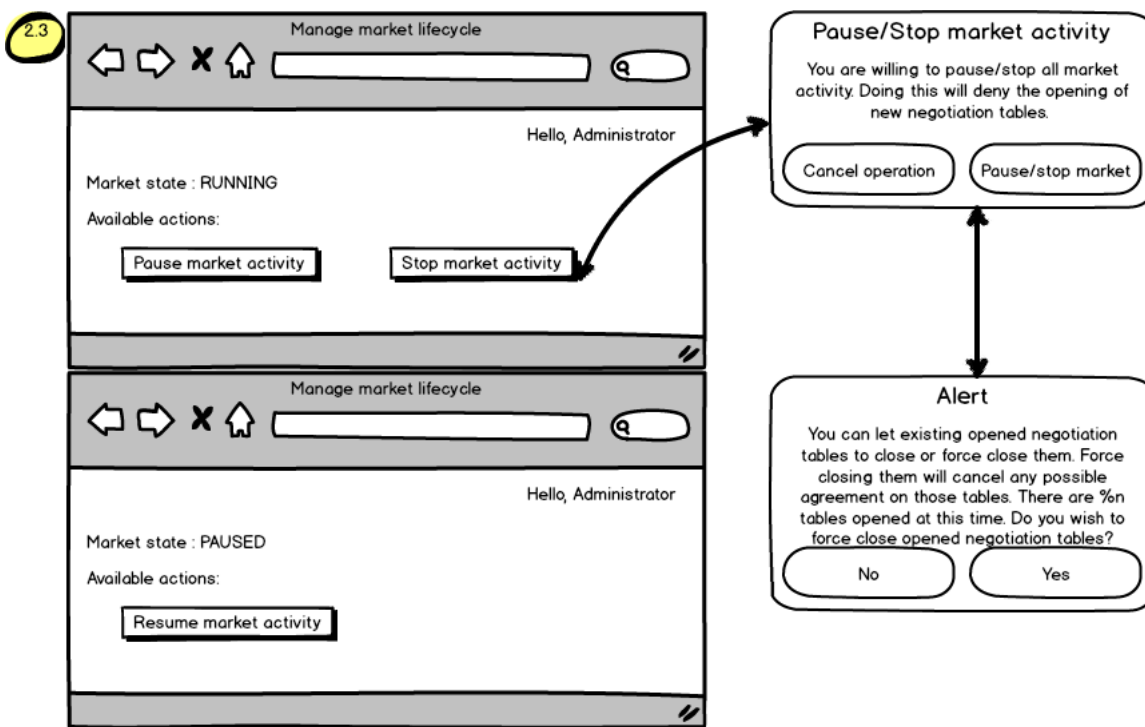


Figura 4.9: Prototipo de la interfaz correspondiente a la gestión del ciclo de vida del mercado

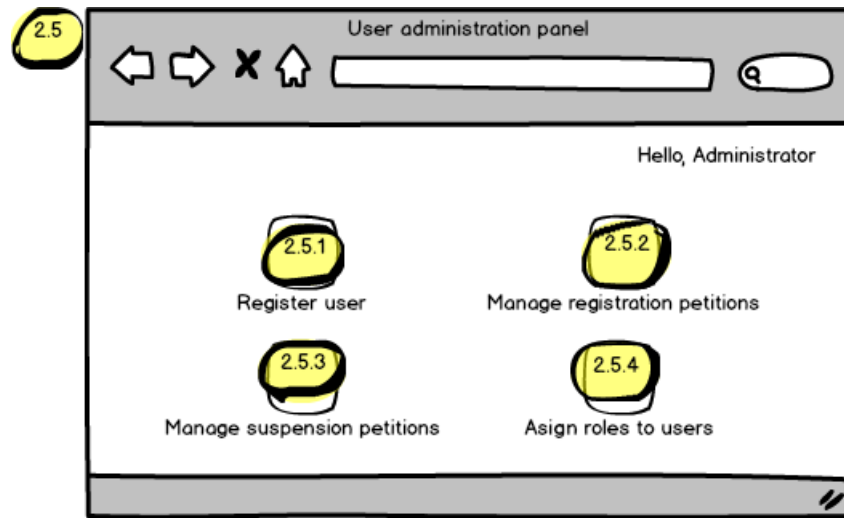


Figura 4.10: Prototipo de la interfaz correspondiente a al panel de administración de usuarios del mercado

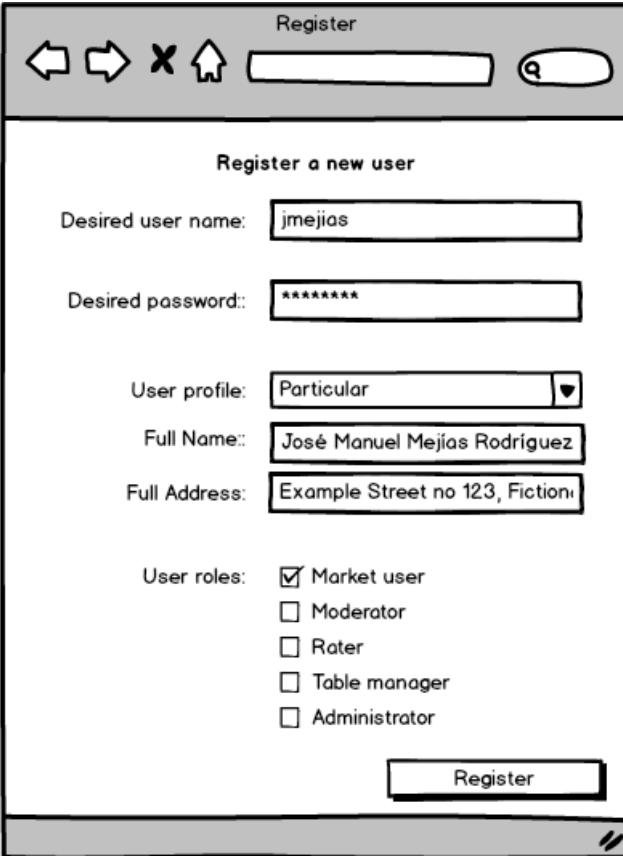
moderadores, etc). La Figura 4.11 representa esta pantalla.

La gestión de las peticiones de registro de los usuarios se controlan desde el panel de gestión de peticiones de registro de usuarios. Tal como se puede ver en la Figura 4.12 se pueden visualizar los datos de cada petición y actuar mediante la aceptación o denegación del registro. En cuanto a las peticiones de suspensión que se podrán emitir por parte de los moderadores, estas se gestionarán en el panel de gestión de peticiones de suspensión (correspondiente a la Figura 4.13).

Por último la gestión de los roles de los distintos usuarios se puede apreciar en la Figura 4.14. En esta página se pueden gestionar los roles que cada usuario dispone y ajustarlos en función de los cargos de responsabilidad que se desean para cada uno.

### Panel de administración de tasaciones

La administración de tasaciones correspondientes al rol tasador dentro de la organización que gestiona el mercado se realiza desde una única página (Figura 4.24). Desde ella se pueden gestionar tanto las tasaciones realizadas como las tasaciones pendientes de su realización. En el panel se puede ver una especificación detallada del producto a tasar y asignar un valor aproximado en base a las propiedades del producto descrito. Además, se puede marcar que la tasación de productos similares



The image shows a hand-drawn prototype of a web interface for manual user registration. The interface is contained within a rectangular frame with a grey header and footer. The header contains the title "Register" and navigation icons: a left arrow, a right arrow, a close (X) icon, a home icon, a search bar with a magnifying glass icon, and a search input field. A yellow speech bubble with the number "2.5.1" is positioned to the left of the header. The main content area is titled "Register a new user" and contains several form fields: "Desired user name:" with the value "jmejias", "Desired password:" with "\*\*\*\*\*", "User profile:" with a dropdown menu showing "Particular", "Full Name:" with "José Manuel Mejias Rodríguez", and "Full Address:" with "Example Street no 123, Fiction". Below these fields is a "User roles:" section with five checkboxes: "Market user" (checked), "Moderator", "Rater", "Table manager", and "Administrator". A "Register" button is located at the bottom right of the form area. The footer contains a small icon of a pen and a slash.

Figura 4.11: Prototipo de la interfaz correspondiente al registro manual de usuarios por parte del gestor de mercado

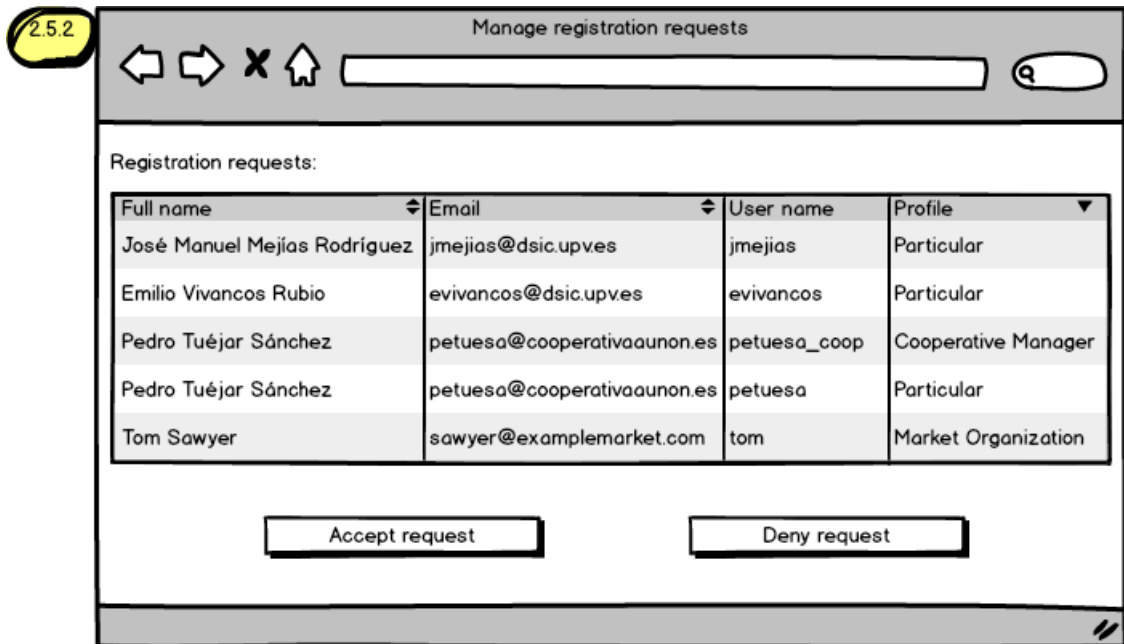


Figura 4.12: Prototipo de la interfaz correspondiente a la gestión de peticiones de registro de usuarios

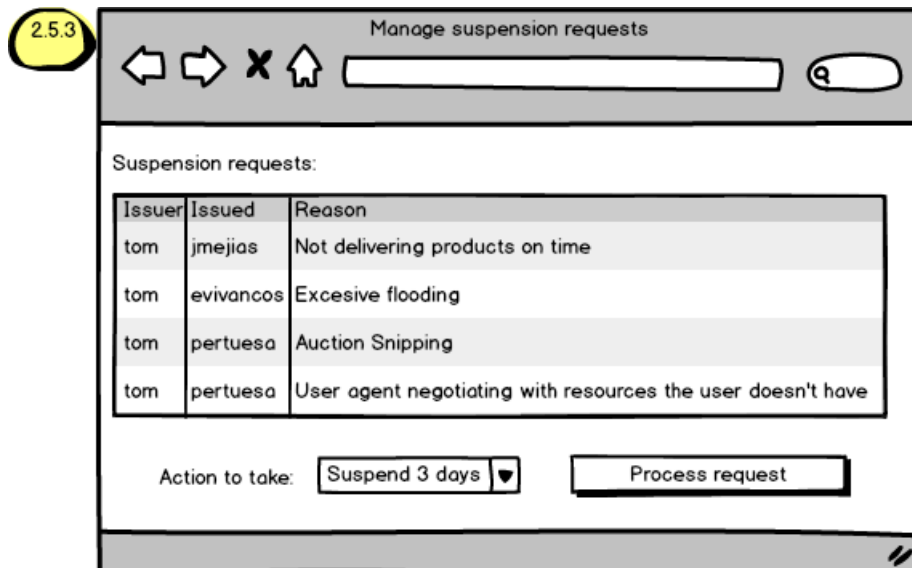


Figura 4.13: Prototipo de la interfaz correspondiente a la visualización de peticiones de suspensión de usuarios

2.5.4

Assign roles to users

User	Market user	Moderator	Rater	Table	Administrator
Administrator	true	true	true	true	true
Moderator	false	true	false	false	false
Rater	false	false	true	false	false
Table manager	false	false	false	true	false
jmejias	true	false	false	false	false

Figura 4.14: Prototipo de la interfaz correspondiente al panel para modificar los roles disponibles para cada usuario del mercado

al tasado se gestionen de forma automática dando el valor de tasación indicado en la primera tasación. Ésto es especialmente útil en mercados en la que los productos no cambian su valor de forma drástica con el paso del tiempo. Por último, un tasador puede revisar y modificar cualquier tasación realizada previamente seleccionándolo de la lista de tasaciones.

### Panel de moderación de conflictos

La moderación de conflictos dentro de la plataforma, acción realizada por los miembros de la organización con el rol *Moderator*, se gestiona desde dos páginas. En la primera, representada en la Figura 4.16, se puede ver las moderaciones pendientes de resolución. Al hacer *click* en ver detalles, se puede observar de forma detallada la descripción del conflicto y las posibilidades de resolución del mismo en base al estado del acuerdo que se ha alcanzado hasta ese punto. Desde ese panel (mostrado en la Figura 4.17) se puede solicitar, además, pedir la suspensión de uno o varios usuarios involucrados en el conflicto.



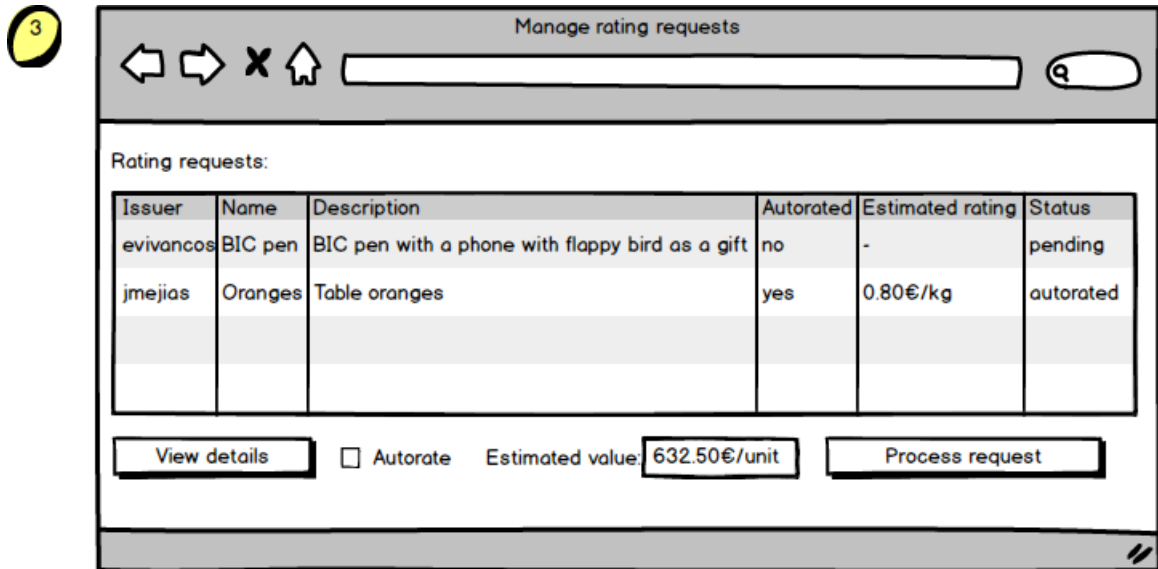


Figura 4.15: Prototipo de la interfaz correspondiente a la visualización y resolución de las peticiones de tasado de productos

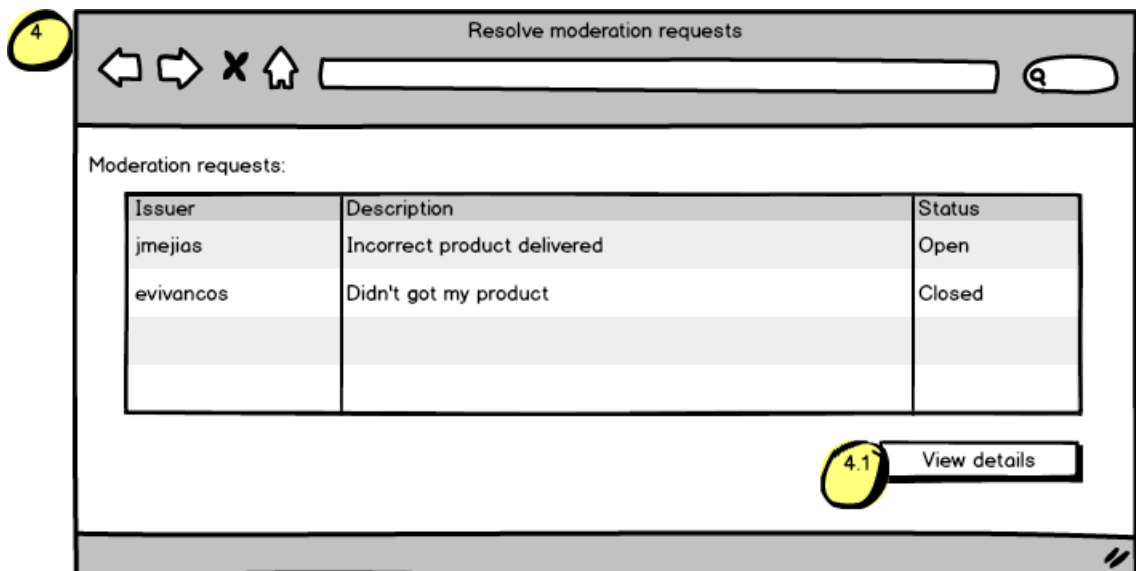


Figura 4.16: Prototipo de la interfaz correspondiente a la visualización de peticiones de moderación de conflictos

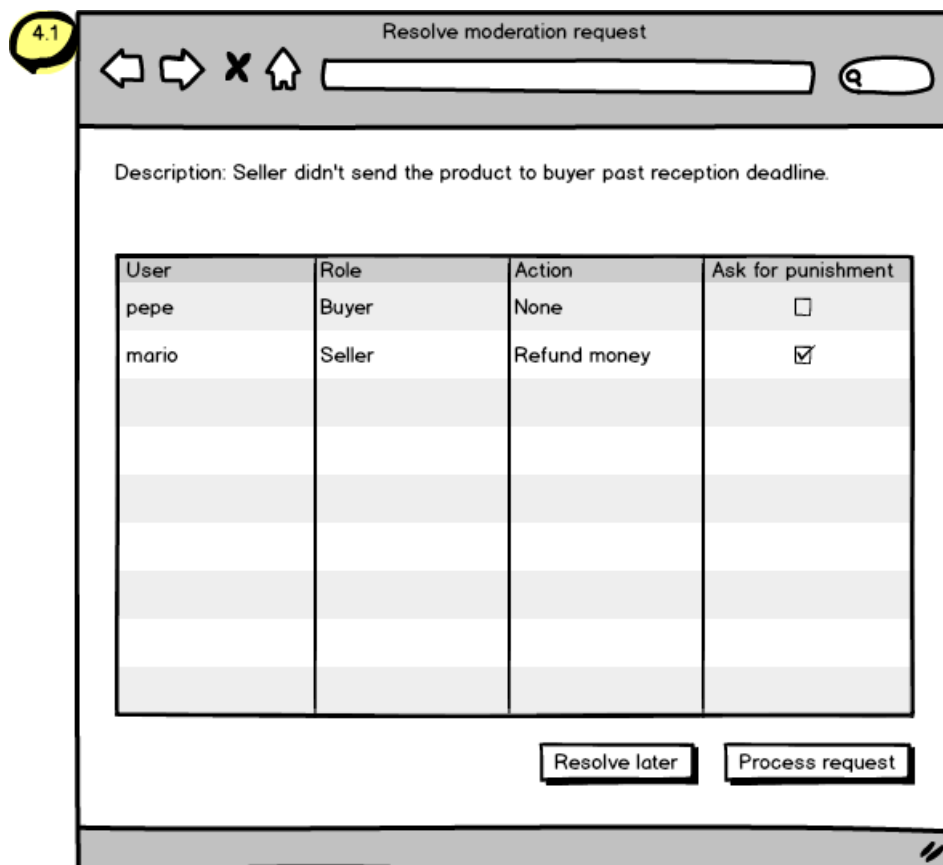


Figura 4.17: Prototipo de la interfaz correspondiente a la resolución de un conflicto por parte de un moderador

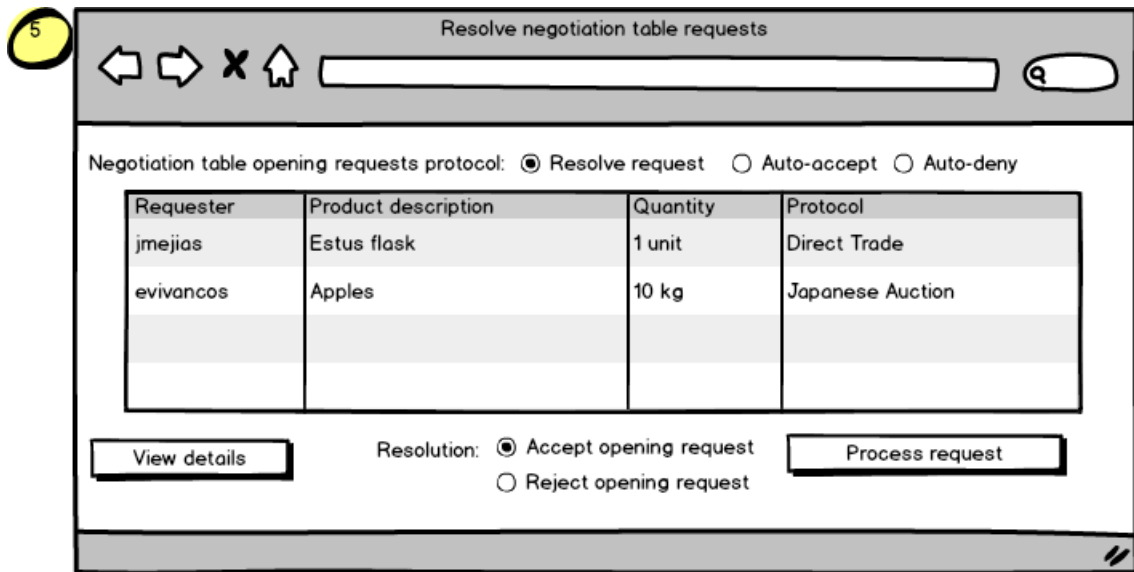


Figura 4.18: Prototipo de la interfaz correspondiente a la gestión de las peticiones de apertura de las mesas de negociación

### Panel de administración de mesas de negociación

La administración de las peticiones de apertura de mesas de negociación para la venta de los diversos productos ofrecidos por la plataforma se puede observar en la Figura 4.18. En esta página se puede observar las peticiones de apertura de mesas de negociación pendientes de su resolución. El usuario puede ver un detalle más exhaustivo de cada petición como el producto o los parámetros de la mesa como el protocolo, el periodo en el que estará abierta, etc. Por último, el usuario puede decidir si aceptar o rechazar cada petición de apertura, notificando al solicitante la resolución de la petición de apertura.

### Uso e interacción de la plataforma de mercado

El uso e interacción de la plataforma de mercado por parte de los usuarios de la plataforma que no pertenezcan a la organización controladora del mercado comienza con el panel de uso del mercado o *User Dashboard* representada en la Figura 4.19. Desde este panel el usuario puede acceder a las distintas partes de la plataforma. La primera de ellas es el panel de información del mercado, el cual es similar al mostrado a los administradores del mercado en la Figura 4.7. La segunda de ellas es la de

mostrar las mesas de negociación disponibles en ese momento (Figura 4.20).

Se pueden observar las mesas de negociación en función del estado de la mesa, el protocolo o buscar entre las mesas disponibles. Si el usuario tienen interés en observar el estado de la mesa de negociación se accede a un panel con más detalles de la misma y de las acciones que puede realizar para negociar dentro de la mesa (Figura 4.22). Por otra parte, el usuario puede solicitar la apertura de una mesa de negociación mediante la definición del producto con el que desea comerciar y sus propiedades (Figura 4.23).

En tercer lugar, el usuario puede solicitar la petición de tasación de un producto. Para ello accede al panel de administración de tasaciones (Figura 4.15) en el que puede observar el estado de las tasaciones realizadas y mediante un panel similar al de la Figura 4.23 puede definir el producto a tasar. Seguidamente, el usuario puede observar el estado de los acuerdos alcanzados dentro de la plataforma (Figura 4.25). En este panel el usuario puede realizar acciones tales como aceptar o rechazar el acuerdo alcanzado o marcar un estado del acuerdo (como por ejemplo debido a haberlo pagado, haber recibido el pago, haber enviado el producto o haberlo recibido).

Por otra parte, el usuario puede manejar el estado del agente que lo representa de forma automática en el mercado (Figura 4.26). En este panel el usuario puede observar los objetivos pendientes de realizar por parte del usuario, añadir más objetivos de compra y venta de productos y ofrecimiento y solicitud de servicios, cambiar el criterio que ha de seguir el agente para alcanzar acuerdos y fijar el presupuesto máximo del que dispone.

Por último, el usuario puede cambiar algunos de sus atributos como usuario de la plataforma mediante un formulario (mostrado en la Figura 4.27).

## 4.6. Prototipo

En esta sección se detalla el diseño y desarrollo del prototipo que plantea un subconjunto de las funciones de la plataforma de mercado analizada y diseñada en los capítulos anteriores. Para ello se aplican las tecnologías del acuerdo y los sistemas multiagente para ofrecer la funcionalidad que se expondrá más adelante.

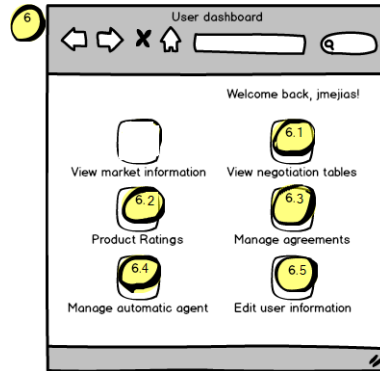


Figura 4.19: Prototipo de la interfaz correspondiente al panel de uso del mercado a un usuario

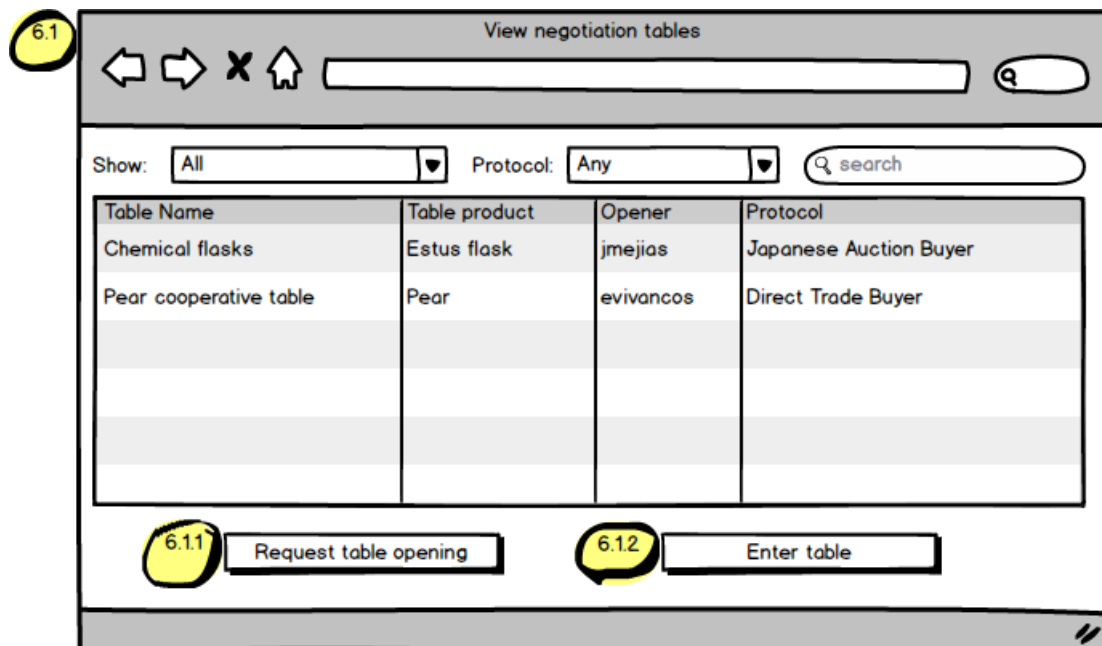


Figura 4.20: Prototipo de la interfaz correspondiente a la visualización de las mesas de negociación del mercado

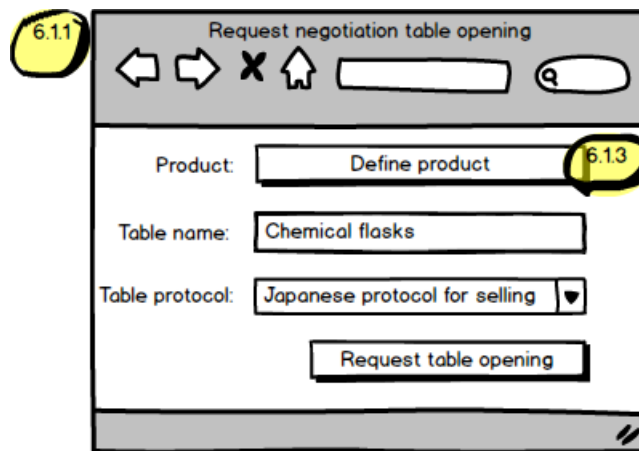


Figura 4.21: Prototipo de la interfaz correspondiente a la petición de apertura de una mesa de negociación

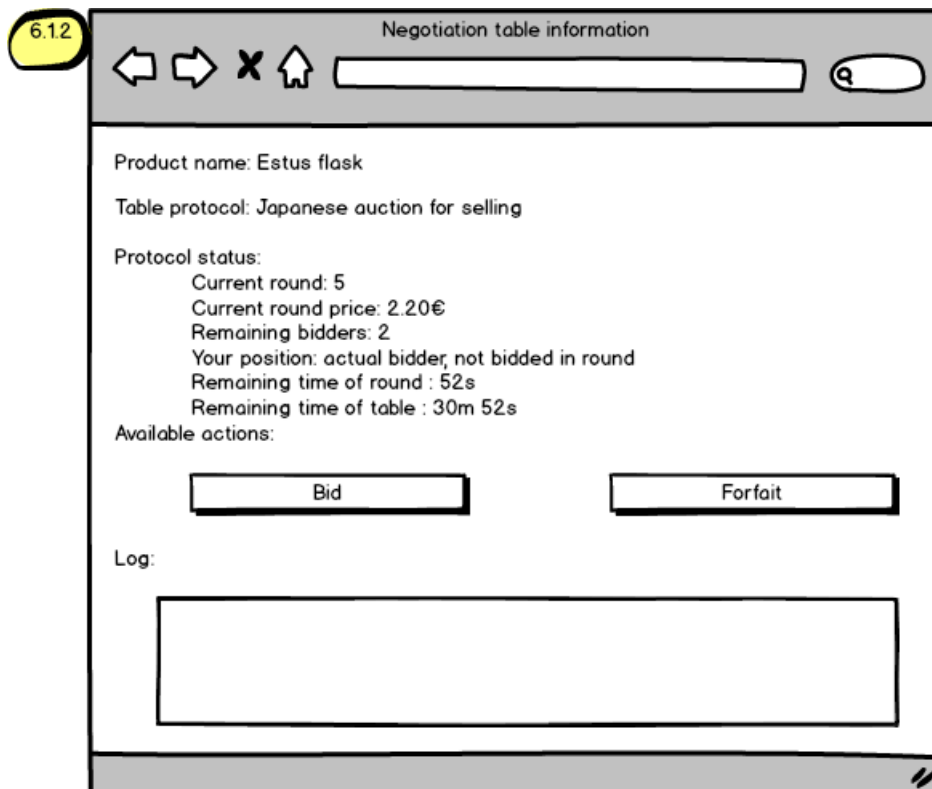


Figura 4.22: Prototipo de la interfaz correspondiente a la visualización de una mesa de negociación y las acciones a realizar

6.13

Define product

← → × ↶ 🔍

Measuring unit:

Property	Value	Measuring Unit
Product type	Estus Flask	type
Quantity	5	unit
Heigth	10	cm
Color	orange	color

Figura 4.23: Prototipo de la interfaz correspondiente a la definición de un producto

6.2

Product ratings

← → × ↶ 🔍

6.2.1

Product name	Product description	Estimated value
BIC pen	BIC pen with a phone with flappy bird as a gift	637.50€/unit
Oranges	Table oranges	0.80€/kg

Figura 4.24: Prototipo de la interfaz correspondiente a la visualización de las peticiones de tasación realizadas

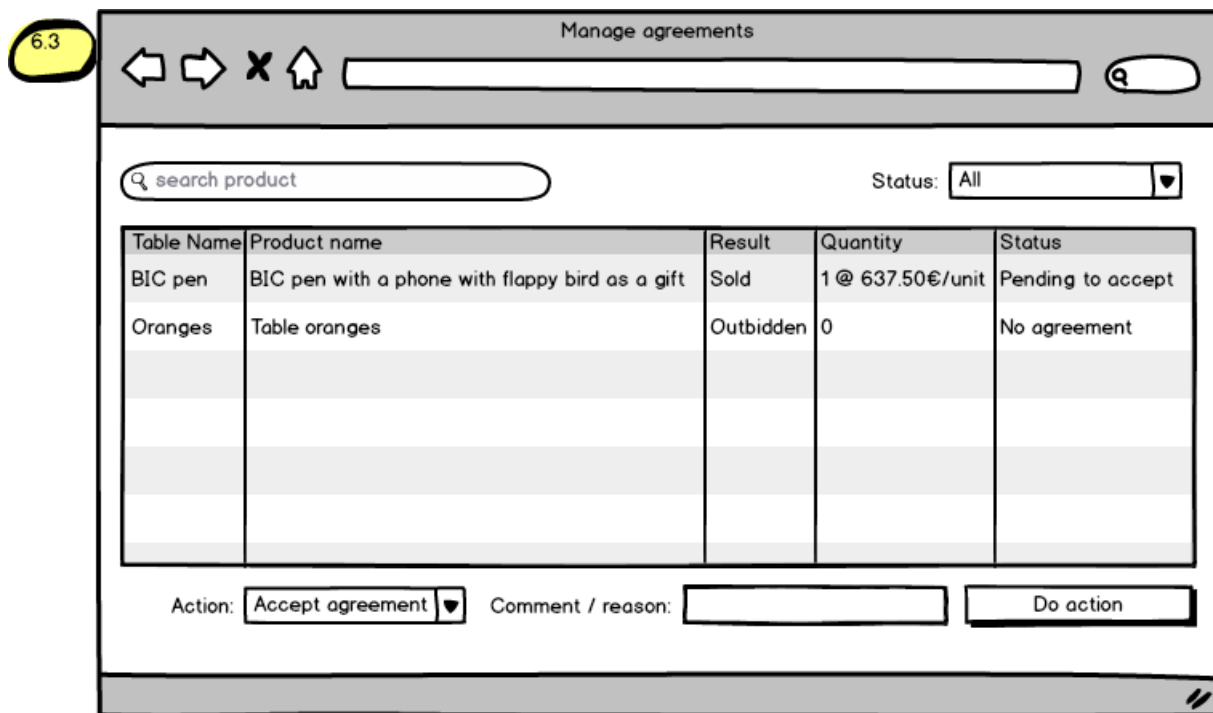


Figura 4.25: Prototipo de la interfaz correspondiente a la gestión de los acuerdos realizados y su estado



6.4

Manage automatic agent

← → × ↶

Search

Current objectives:

Action Type	Product	Quantity
Buy	Table Oranges	100 kg
Sell	Orange Juice Oranges	250 kg

View product information Remove objective

Add objective:

Action: Buy

Product: Define product

Add objective

Agent's utility criteria: Social benefit

Agent's maximum budget: 385.25€

Figura 4.26: Prototipo de la interfaz correspondiente a la gestión del agente representante del usuario

6.5

Edit user information

← → ✕ 🏠 🔍

Old password:

Desired password:

User profile:

Full Name:

Full Address:

Figura 4.27: Prototipo de la interfaz correspondiente a la modificación de los datos del usuario

#### 4.6.1. Diseño del prototipo

El prototipo consta de la demostración del uso de las capacidades del mercado en un escenario de uso típico en el que se involucran tanto un comprador y un vendedor a la hora de alcanzar un acuerdo sobre la comercialización directa de un producto con un precio acordado. Para ello se involucran también las partes del mercado que mantienen su estructura y formalizan los acuerdos y procesos derivados de los mismos.

Como base partimos de la existencia de usuarios del mercado en los que pueden ser tanto vendedores como compradores. Estos usuarios se registran en el mercado para posteriormente identificarse y entrar en el mismo introduciendo sus credenciales de acceso. Dentro del mercado los usuarios compradores disponen de la posibilidad de listar productos con un precio de venta determinado. Para ello se procede a determinar las condiciones de venta y la descripción del producto. Después el posible comprador puede listar las mesas de negociación (productos) abiertas, ver los detalles de cada una de las mesas y proceder a comprar el producto si lo desea. Finalmente, el usuario puede observar el resultado de la compra en base al acuerdo formalizado.

Dentro de este proceso de implementación se emplean las producciones de la metodología ROMAS detalladas en el Capítulo 3. Es por ello por lo que representamos

a los agentes involucrados como miembros del sistema multiagente que constituye el mercado con roles definidos en la metodología. De ésta forma, consideramos los roles *Market User*, *Table Controller* y *Market Manager* como agentes que representan los usuarios, el administrador de las mesas de negociación y el gestor y controlador del mercado respectivamente. De esta forma podemos considerar que cada usuario es representado por un agente con el rol *Market User*, un agente gestionará las mesas de negociación y su funcionamiento como *Table Controller* y toda la funcionalidad relacionada con la gestión del prototipo de mercado se gestiona a partir de un agente con el rol *Market Manager*. A continuación se detallarán las características de cada uno de estos roles.

### **Market User**

Los usuarios de la plataforma de mercado (representados por el rol *Market User*) son los agentes que interactúan con el mercado con el objetivo de alcanzar acuerdos dentro de la plataforma de mercado. Tanto los compradores como los vendedores del mercado disponen de este rol. Cada agente representa a un único usuario. Múltiples agentes pueden representar a un usuario. El usuario no dispone de acceso a la capa de persistencia de la plataforma por lo que requiere realizar peticiones a otros agentes del sistema para acceder a la información que requieran. El agente puede gestionarse de forma controlada por el usuario directamente mediante el uso de una interfaz o de forma automática en base a sus objetivos prefijados. Cada usuario gestiona sus credenciales mediante un par usuario/contraseña definidas en el momento de registro.

### **Market Manager**

El controlador de la plataforma de mercado (representado por el rol *Market Manager*) es el agente o conjunto de agentes que se encargan de gestionar el funcionamiento del mercado, así como su ciclo de vida y transacciones. En el caso de haber una alta carga de transacciones en el mercado, el rol puede ser empleado por múltiples agentes en cuyo caso su carga se vería distribuida por la plataforma de gestión de agentes. El agente gestiona las peticiones realizadas de forma automática y sin la necesidad del control por parte de los usuarios. Por otra parte el agente puede realizar acciones por su propia iniciativa mediante activadores para tareas de mantenimiento de la plata-

forma, tales como el registro en lotes de las transacciones con el objetivo de reducir la carga.

### Table Controller

El controlador de mesas del mercado (representado por el rol *Table Controller*) es el agente o conjunto de agentes que se encargan de gestionar el funcionamiento de las mesas del mercado. En el caso de haber una alta carga de transacciones en el mercado y tal como ocurre con el *Market Manager*, el rol puede ser empleado por múltiples agentes. El agente gestiona las peticiones de apertura y listado de las mesas de negociación así como el ciclo de vida de las mismas. Al igual que el controlador del mercado el agente puede realizar por iniciativa propia acciones relacionadas con las mesas de negociación.

#### 4.6.2. Implementación del prototipo

La implementación del prototipo se ha realizado empleando las herramientas anteriormente descritas. En primer lugar se ha creado una base de datos que contenga el modelo de datos del diagrama entidad-relación especificado en la Sección 4.5.1. Para ello se empleó la herramienta SQLWorkbench y una base de datos local MySQL. Seguidamente, se implementó el puente entre el modelo de datos de la capa de persistencia y los objetos de acceso a datos (*Data Access Objects* o DAO) de la capa lógica mediante el uso de Hibernate en el IDE empleado (IntelliJ Idea). De esta forma y como se puede observar en la Figura 4.28 se ha generado de forma automática esta capa lógica. Cada uno de los ficheros contiene una clase Java que se encarga de gestionar una de las entidades diseñadas anteriormente. Como ejemplo de ello se puede observar en la Figura 4.29 el código generado automáticamente para la entidad Usuario.

Tras crear las facilidades que permiten gestionar la comunicación entre la capa lógica y la capa de persistencia se procedió a implementar los agentes que forman la plataforma de mercado. Para ello se comenzó implementado el controlador de mercado o *Market Manager*. Para ello se empleó el framework MAGENTIX2 (visto en la Sección 4.3). Parte de su implementación se puede observar en la Figura 4.30. Se partió de un agente base de la plataforma para implementar su funcionalidad. Para adaptarlo a nuestras necesidades se ha sobrescrito el método *execute*. Su funciona-

```
CategoryEntity.java
JapaneseAuctionEntity.java
JapaneseAuctionEntityPK.java
MarketAgreementStatusEntity.java
MarketAgreementStatusEntityPK.java
MarketEntity.java
MarketStatisticEntity.java
MarketStatisticEntityPK.java
MeasuringUnitEntity.java
ModerationRequestEntity.java
ModerationRequestEntityPK.java
NegotiationTableEntity.java
NegotiationTableEntityPK.java
ObjectiveEntity.java
ObjectiveEntityPK.java
OpeningRequestEntity.java
OpeningRequestEntityPK.java
ProductEntity.java
ProductEntityPK.java
ProfileEntity.java
ProfileEntityPK.java
PropertyEntity.java
PropertyEntityPK.java
ProtocolActionEntity.java
ProtocolActionEntityPK.java
ProtocolEntity.java
RatingRequestEntity.java
RatingRequestEntityPK.java
SuspensionEntity.java
SuspensionEntityPK.java
TradingHallEntity.java
TradingHallEntityPK.java
UserAgentEntity.java
UserAgentEntityPK.java
UserEntity.java
```

Figura 4.28: Listado de clases generadas mediante Hibernate

```
package genericmarket.model;

import javax.persistence.*;

@Entity
@Table(name = "User", schema = "", catalog = "genericmarket")
@IdClass(UserEntityPK.class)
public class UserEntity {
    private int iduser;
    private String username;
    private String password;
    private String fullname;
    private String address;
    private String email;
    private int idUser;
    private String fullName;
    private int marketIdMarket;

    @Id
    @Column(name = "iduser")
    public int getIduser() {
        return iduser;
    }

    public void setIduser(int iduser) {
        this.iduser = iduser;
    }

    @Basic
    @Column(name = "username")
    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    @Basic
    @Column(name = "password")
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    @Basic
    @Column(name = "fullname")
    public String getFullname() {
        return fullname;
    }
}
```

Figura 4.29: Código generado de forma automática para la entidad Usuario

miento consiste en ir recibiendo mensajes con peticiones de forma indefinida durante la ejecución del mercado. Estos mensajes serán procesados por el agente y devolverán una respuesta con el resultado. En el extracto de código se muestra el agente gestiona los mensajes recibidos del agente para resolver peticiones de identificación, registro y cambio de contraseña. Al solicitar una operación que requiera un acceso a la base de datos se emplean los objetos DAO generados de forma automática para gestionar el almacenamiento en la base de datos de la capa de persistencia. Este proceso en el caso del registro de nuevos usuarios se puede visualizar en la Figura 4.31.

Seguidamente, se implementó la funcionalidad del agente con el rol *Table Manager*. Para ello se implementó mediante la extensión del agente base del framework una clase Java que incluya su funcionalidad. Parte de su implementación se puede observar en la Figura 4.32. Su labor es la de recibir mensajes conteniendo peticiones relacionadas con las mesas de negociación y responderlas.

Por último se implementó la funcionalidad del agente con el rol *Market User*. Para ello se implementó una clase Java de forma similar a los casos anteriores. En la Figura 4.33 se observan los métodos implementados. Su labor es la de interactuar con el usuario físico y comunicarse con los agentes de la plataforma para realizar peticiones tanto las solicitadas por el usuario como las realizadas de forma automática. En este caso y por simplicidad la interacción entre el usuario físico y el agente se realiza mediante terminal. En la Figura 4.34 se puede observar parte del código que realiza esta interacción. Las peticiones realizadas sobre otros agentes las realiza haciendo peticiones mediante el envío de mensajes. En la Figura 4.35, por ejemplo, se realiza una petición desde el agente que representa el usuario al administrador del mercado para registrarse en la plataforma.

El resultado de la implementación se puede observar en las Figuras 4.36 y 4.37 en la que se muestra la interacción con el usuario para su registro, identificación, apertura de una mesa de negociación y compra de un producto.

En este punto finaliza la implementación del prototipo. Se ha creado una aplicación de escritorio controlada mediante terminal que permite la interacción con un mercado para la compraventa de productos. En el próximo capítulo se mencionan las conclusiones alcanzadas y el trabajo futuro a realizar.

```

package genericmarket.prototype;

import es.upv.dsic.gti_ia.core.ACLMessage;
import es.upv.dsic.gti_ia.core.AgentID;
import es.upv.dsic.gti_ia.core.SingleAgent;

import java.util.HashMap;

public class MarketManager extends SingleAgent {

    private HashMap<String, String> userHash;
    private AgentID tableManagerID;

    public MarketManager(AgentID id, AgentID tableManagerID) throws Exception {
    }

    public void execute() {
        try {
            while (true) {
                ACLMessage msg = receiveACLMessage();
                String[] petition = msg.getContent().split(":");
                String requestType = petition[0];
                if (requestType.equals("login")) {
                    ACLMessage response = new ACLMessage();
                    response.setReceiver(msg.getSender());
                    response.setSender(this.getAid());
                    if (checkLoginCredentials(petition[1], petition[2])) {
                        response.setContent("OK");
                        response.setPerformative(ACLMessage.AGREE);
                        //TODO this should return a partial credential
                    } else {
                        response.setContent("KO");
                        response.setPerformative(ACLMessage.REFUSE);
                    }
                    send(response);
                } else if (requestType.equals("register")) {
                    ACLMessage response = new ACLMessage();
                    response.setReceiver(msg.getSender());
                    response.setSender(this.getAid());
                    String registerResult = registerUser(petition[1], petition[2]);
                    if (registerResult.equals("OK")) {
                        response.setContent("OK");
                        response.setPerformative(ACLMessage.AGREE);
                    } else {
                        response.setContent(registerResult);
                        response.setPerformative(ACLMessage.REFUSE);
                    }
                    send(response);
                } else if (requestType.equals("changepassword")) {
                    ACLMessage response = new ACLMessage();
                    response.setReceiver(msg.getSender());

```

Figura 4.30: Extracto de código correspondiente al *Market Manager*



```
private static boolean registerUserInDatabase(String username, String password, String address, String email, String fullname) {  
    try{  
        UserEntity insertUser = new UserEntity();  
        insertUser.setAddress(address);  
        insertUser.setEmail(email);  
        insertUser.setFullname(fullname);  
        insertUser.setPassword(password);  
        insertUser.setUsername(username);  
  
        SessionFactory sessFact = HibernateUtil.getSessionFactory();  
        Session session = sessFact.openSession();  
  
        session.beginTransaction();  
  
        session.save(insertUser);  
  
        session.getTransaction().commit();  
  
        return true;  
    }catch (Exception e){  
        e.printStackTrace();  
        return false;  
    }  
}
```

Figura 4.31: Extracto de código correspondiente al registro de un usuario en la base de datos

```

public class TableManager extends BaseAgent {

    public HashMap<String,Table> tables;
    public TableManager(AgentID id) throws Exception {
        super(id);
        tables = new HashMap<String, Table>();
    }

    public void execute() {

        try {
            while(true) {
                ACLMessage msg = receiveACLMessage();
                String[] petition = msg.getContent().split(":");
                String requestType = petition[0];
                if(requestType.equals("createtable")){
                    ACLMessage response = new ACLMessage();
                    response.setReceiver(msg.getSender());
                    response.setSender(this.getAid());
                    String createTableResult = createTable(petition[1], petition[2], petition[3], petition[4], petition[5]);
                    if (createTableResult.equals("OK")) {
                        response.setContent("OK");
                        response.setPerformative(ACLMessage.AGREE);
                    } else {
                        response.setContent(createTableResult);
                        response.setPerformative(ACLMessage.REFUSE);
                    }
                }
                send(response);
            }
            else if(requestType.equals("requesttables")){
                ACLMessage response = new ACLMessage();
                response.setReceiver(msg.getSender());
                response.setSender(this.getAid());
                Set<String> tableSet = tables.keySet();
                response.setPerformative(ACLMessage.AGREE);
                String result = "";
                for (String table : tableSet){
                    result += table + "\n";
                }
                response.setContent(result);
                send(response);
            }
            else if(requestType.equals("requesttabledetail")){
                ACLMessage response = new ACLMessage();
                response.setReceiver(msg.getSender());
                response.setSender(this.getAid());
                response.setPerformative(ACLMessage.AGREE);
                String tableDetails = tables.get(petition[1]).toString();
                response.setContent(tableDetails);
                send(response);
            }
        }
    }
}

```

Figura 4.32: Extracto de código correspondiente al *Table Controller*

```
public void execute() throws InterruptedException {  
}  
private void interactWithUser() throws InterruptedException {  
}  
private void userPasswordChange() throws InterruptedException {  
}  
private void userSignUp() throws InterruptedException {  
}  
private void userLogin() throws InterruptedException {  
}  
private void postLoginScreen() throws InterruptedException {  
}  
private void listNegotiationTables() throws InterruptedException {  
}  
private void getTableDetails(int option) throws InterruptedException {  
}  
private int getNegotiationTableList() throws InterruptedException {  
}  
private int getNegotiationTableInfo() throws InterruptedException {  
}  
private void createNegotiationTable() throws InterruptedException {  
}  
private void createListing(String title, String description, Float sellingPrice) throws Int  
}  
private void negotiateOnDirectTrade(){  
}  
private void negotiateOnDirectTradeAuto(){  
}  
private void negotiateOnDirectTradeSeller(){  
}  
private void listUserAgreements(){  
}  
public boolean isAlive() {  
}
```

Figura 4.33: Listado de métodos implementados en la clase MarketUser.java

```
public void execute() throws InterruptedException {
    System.out.println("Executing, I'm " + getName() + " and I'm a market user");
    System.out.println("Welcome to prototype market!");
    interactWithUser();
    System.out.println("Market user out."); //TODO it never ends?
    alive = false;
}

private void interactWithUser() throws InterruptedException {
    boolean quit = false;
    do {
        System.out.println("Select an option:");
        System.out.println("1. Log in");
        System.out.println("2. Sign up");
        System.out.println("3. Remember password");
        System.out.println("q. Quit");
        String optionLine;
        do {
            optionLine = kbd.nextLine();
        } while (optionLine == null || optionLine.equals(""));
        char option = optionLine.charAt(0);

        switch (option) {
            case '1': //log in
                userLogin();
                break;
            case '2': //Sign up
                userSignUp();
                break;
            case '3': //Remember password
                userPasswordChange();
                break;
            case 'Q':
            case 'q':
                System.out.println("Goodbye!");
                quit = true;
                break;
            default:
                System.out.println("Incorrect option, try again");
        }
    } while (!quit);
}
```

Figura 4.34: Extracto de código que representa parte la interacción con el usuario

```
private void userSignUp() throws InterruptedException {
    System.out.println("Please enter your credentials for signing up the market");
    System.out.print("Username: ");
    username = kbd.nextLine();

    Console console = System.console();
    String password;
    if (console != null) {
        char passwordArray[] = console.readPassword("Password: ");
        password = String.valueOf(passwordArray);
    } else {
        System.out.print("Password: ");
        password = kbd.nextLine();
    }
    ACLMessage message = new ACLMessage();
    message.setReceiver(marketManager); //TODO add search on organization
    message.setSender(this.getAid());
    message.setPerformative(ACLMessage.REQUEST);
    message.setContent("register:" + username + ":" + password);
    send(message);
    ACLMessage response = receiveACLMessage();
    if (response.getPerformative().equals("AGREE")) {
        System.out.println("Register OK! Now you can log in with your credentials");
    } else {
        System.out.print("Register KO! ");
        System.out.println(response.getContent());
    }
}
```

Figura 4.35: Extracto de código que representa la interacción con el administrador de mercado para el registro de un usuario

```
Select an option:
1. Log in
2. Sign up
3. Remember password
q. Quit
2
Please enter your credentials for signing up the market
Username: testbuyer
Password: testpassword
Select an option:
1. Log in
2. Sign up
3. Remember password
q. Quit
2
Please enter your credentials for signing up the market
Username: testseller
Password: testpassword
Register OK! Now you can log in with your credentials
Select an option:
1. Log in
2. Sign up
3. Remember password
q. Quit
1
Please enter your credentials for entering the market
Username: testseller
Password: testpassword
Login OK!
Select an option:
1. Create a new negotiation table
2. List negotiation tables
3. List agreements
q. Logout
1
Select a protocol:
1. Direct selling
q. Quit
1
Please pick a title for your listing: Example product
Please describe your product: This is an example product
Introduce the price to sell the product (in euros): 123,45
Product listing OK!
Select an option:
1. Create a new negotiation table
2. List negotiation tables
3. List agreements
q. Logout
q
```

Figura 4.36: Interacción entre el usuario y el sistema multiagente

```
Select an option:
1. Log in
2. Sign up
3. Remember password
q. Quit
.....
1
Please enter your credentials for entering the market
Username: testbuyer
Password: testpassword
Login OK!
Select an option:
1. Create a new negotiation table
2. List negotiation tables
3. List agreements
q. Logout
2
Available negotiation tables:
1. Example product
Please enter the number of table to enter or enter q to quit: 1
Negotiation table details:
Table title: Example product
Product description: This is an example product
Product price (in euros): 123,45
Available actions:
1. Buy
q. Quit
Please enter an action: 1
Product bought. An agreement has been met.
Select an option:
1. Create a new negotiation table
2. List negotiation tables
3. List agreements
q. Logout
3
User agreements:
1. This is an example product. Price: 123,45 euros
Press ENTER to continue
```

Figura 4.37: Interacción entre el usuario y el sistema multiagente (continuación)

# Capítulo 5

## Conclusiones y trabajo futuro

En este capítulo se exponen las aportaciones que han generado las distintas herramientas empleadas al proyecto, las conclusiones que se han alcanzado y el trabajo futuro propuesto a partir de los resultados obtenidos.

### 5.1. Conclusiones

El resultado del trabajo realizado es un diseño completo y estructurado de un sistema multiagente y de la organización que compondrá una plataforma de mercado en la que los usuarios podrán vender y comprar productos y ofrecer y solicitar servicios. Desde este punto de vista se ha diseñado un modelo de negocio orientado a esta plataforma de mercado similar al existente en la actualidad y que pese a su explotación por parte de otras empresas tales como Amazon y Ebay, es un fenómeno en auge con la crecida de tendencias como el *Big Data* y los sistemas multiagente en la industria.

En cuanto al uso de la metodología ROMAS cabe concluir que sus aportaciones han permitido estructurar y diseñar de forma eficiente el sistema desde el uso del paradigma de los sistemas multiagente y las tecnologías del acuerdo. Su uso ha permitido diseñar a los agentes de una forma sencilla y completa en base a los requisitos de la plataforma que se ha pretendido crear. Por otra parte, las distintas fases han creado documentación útil para futuras fases de desarrollo de la plataforma. Por último, el uso de la metodología ha permitido detectar fallos importantes en los requisitos



propuestos, reforzando la validez del diseño final generado su uso.

Se han mostrado funcionalidades basadas en el uso de los sistemas multiagente para crear una interacción entre los agentes para alcanzar acuerdos sobre productos determinados en un protocolo sencillo como es la venta directa. Su desarrollo ha permitido mostrar la capacidad del framework MAGENTIX2 para la gestión de los agentes dentro de la plataforma y de los servicios que ésta ofrece como es el intercambio de mensajes entre agentes.

Finalmente, y a título personal, cabe destacar la capacidad de la plataforma diseñada como herramienta para la investigación en los campos referidos a los sistemas multiagente, las tecnologías del acuerdo y las plataformas orientadas a servicios (*Service Oriented Architectures*, SOA) para facilitar el desarrollo de plataformas automatizadas que permitan a sociedades, colectivos y personas individuales facilitar la adquisición de productos y la contratación de servicios de una forma sencilla, automatizada y con una reducción tanto en costes, tiempo y esfuerzo permitiendo la competitividad entre empresas e individuos para mejorar como sociedad.

## 5.2. Trabajo futuro

A raíz de los resultados dados, el trabajo futuro a partir de este trabajo consiste en primer lugar y a corto plazo en el desarrollo de una plataforma web que permita mejorar la interacción entre el usuario y el sistema multiagente que gestione la plataforma.

Por otra parte, se plantea dar más capacidad de autonomía a los agentes, implementando mecanismos de razonamiento más complejos empleando una arquitectura BDI más robusta (tal como el uso del lenguaje JASON). También se plantea incrementar la personalización de los mercados creados incrementando las capacidades del diseñador del mercado. Más a largo plazo y con la intención de mejorar las capacidades de los agentes automáticos se plantea emplear el uso de algoritmos de búsqueda para la mejora de la capacidad negociadora de los mismos empleando algoritmos evolutivos.

# Bibliografía

- [1] Jorge Agüero, Miguel Rebollo, Carlos Carrascosa, and Vicente Julián. Developing intelligent agents on the android platform. In *Sixth European Workshop on Multi-Agent Systems (EUMAS 2008) pp*, pages 1–14, 2008.
- [2] Juan M Alberola, Jose M Such, Agustin Espinosa, Vicent Botti, and Ana Garcia-Fornes. Magentix: a multiagent platform integrated in linux. In *Proceedings of the Sixth European Workshop on Multi-Agent Systems (EUMAS-2008)*, 2008.
- [3] Bexy Alfonso, Vicente Botti, Antonio Garrido, and Adriana Giret. A mas-based infrastructure for negotiation and its application to a water-right market. *Information Systems Frontiers*, 16(2):183–199, 2014.
- [4] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing multi-agent systems with JADE*, volume 7. John Wiley & Sons, 2007.
- [5] Botti, V. On the design of mWater: a case study for agreement technologies.
- [6] Massimo Cossentino, Vincent Hilaire, Ambra Molesini, and Valeria Seidita. The IEEE-FIPA standard on the design process documentation template. In *Handbook on Agent-Oriented Design Processes*, page 7–17. Springer, 2014.
- [7] Ana Mas et al. *Agentes Software y Sistemas Multi-agente: Conceptos, Arquitecturas y Aplicaciones*. Pearson.
- [8] Emilia Garcia, A. Giret, and V. Botti. A Model-Driven CASE tool for Developing and Verifying Regulated Open MAS. *Science of Computer Programming - Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex*, 78(6):695–704, 2011.

- 
- [9] Emilia Garcia, A. Giret, and V. Botti. *ROMAS methodology. Handbook on Agent-Oriented Design Processes*. Springer Berlin Heidelberg, 2014.
- [10] Olivier Gutknecht and Jacques Ferber. Madkit: A generic multi-agent platform. In *Proceedings of the fourth international conference on Autonomous agents*, pages 78–79. ACM, 2000.
- [11] Vijay Krishna. *Auction theory*. Academic press, 2009.
- [12] Luís Nogueira and Eugénio Oliveira. A multi-agent system for e-insurance brokering. In *Agent Technologies, Infrastructures, Tools, and Applications for e-Services*, pages 263–282. Springer, 2003.
- [13] Sascha Ossowski. *Agreement Technologies*, volume 8. Springer, 2012.
- [14] A. Osterwalder and Y. Pigneur. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley Desktop Editions. Wiley, 2010.
- [15] Alexander Osterwalder's. Business model canvas. 2008.
- [16] Martin Reck. Types of electronic auctions. In *Information and Communications Technologies in Tourism*, pages 236–243. Springer, 1994.
- [17] Pierre-Michel Ricordel and Yves Demazeau. From analysis to deployment: A multi-agent platform survey. In *Engineering societies in the agents world*, pages 93–105. Springer, 2000.
- [18] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.
- [19] Lena Sanders, Denise Pumain, H elene Mathian, France Gu erin-Pace, and Stephane Bura. Simpop: a multiagent system for the study of urbanism. *Environment and Planning B*, 24:287–306, 1997.
- [20] S Sandri et al. mwater: a sandbox for agreement technologies. In *Artificial Intelligence Research and Development: Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence*, volume 202, page 252. IOS Press, 2009.

- 
- [21] R Keith Sawyer. Artificial societies multiagent systems and the micro-macro link in sociological theory. *Sociological Methods & Research*, 31(3):325–363, 2003.
- [22] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2009.
- [23] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.
- [24] Steve Vinoski. Advanced message queuing protocol. *IEEE Internet Computing*, 10(6):87–89, 2006.
- [25] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [26] Franco Zambonelli, Nicholas R Jennings, and Michael Wooldridge. Developing multiagent systems: The gaia methodology. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 12(3):317–370, 2003.

# Apéndice

# Apéndice A

## Diagramas ilustrativos del desglose de objetivos en tareas

Las siguientes figuras corresponden a la documentación generada en la Sección 3.2.2.3 del documento referentes al desglose de los objetivos generados en tareas.

Como se puede observar en la Figura A.1, el objetivo *Fomentar reutilización* deriva dos tareas correspondientes a *Almacenar especificación de mercado* y *Recuperar especificación de mercado*. En el caso del sistema propuesto se pueden observar los diagramas de tareas generados desde un punto de vista general en la Figura A.1 en el que se detallan las tareas que después se detallarán en subtareas en las figuras siguientes.

Las tareas relacionadas con la administración de los usuarios en la Figura A.2 desarrolla la parte de gestión por parte de los organizadores del mercado la administración y control de usuarios, sus roles y sus acciones.

La Figura A.3 detalla las tareas relacionadas con la administración y gestión del mercado incluyendo el ciclo de vida del mismo y las actividades de las distintas entidades de la organización del mercado tales como la tasación y la moderación de conflictos.

En la Figura A.4, finalmente, se detallan las tareas relacionadas con la interacción con el mercado incluyendo todo el proceso de compraventa por parte de los usuarios del mercado y la solicitud de servicios adicionales como la tasación y moderación de conflictos.

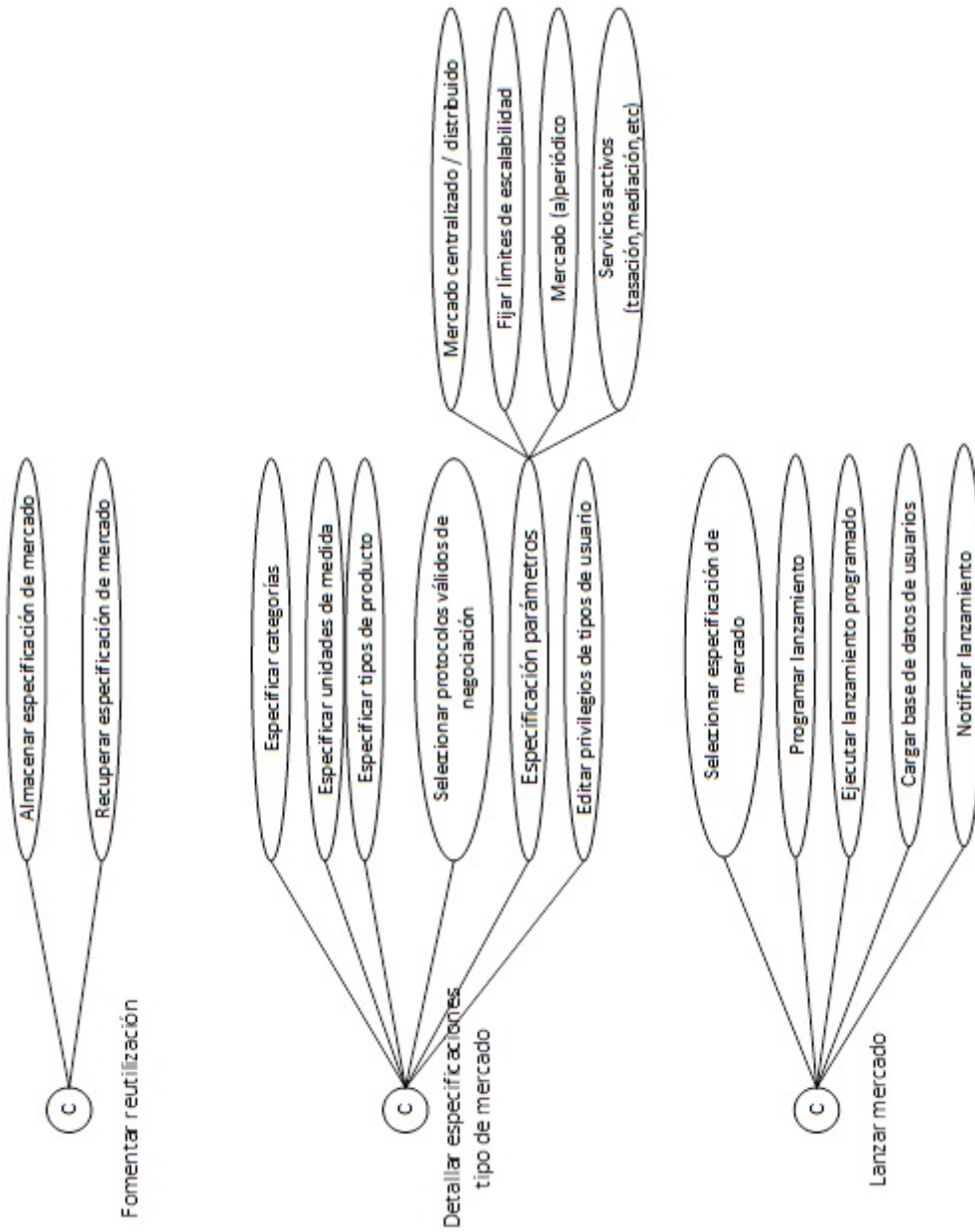


Figura A.1: Diagrama de tareas referente a la definición del mercado



Figura A.2: Diagrama de tareas referente a la administración de usuarios

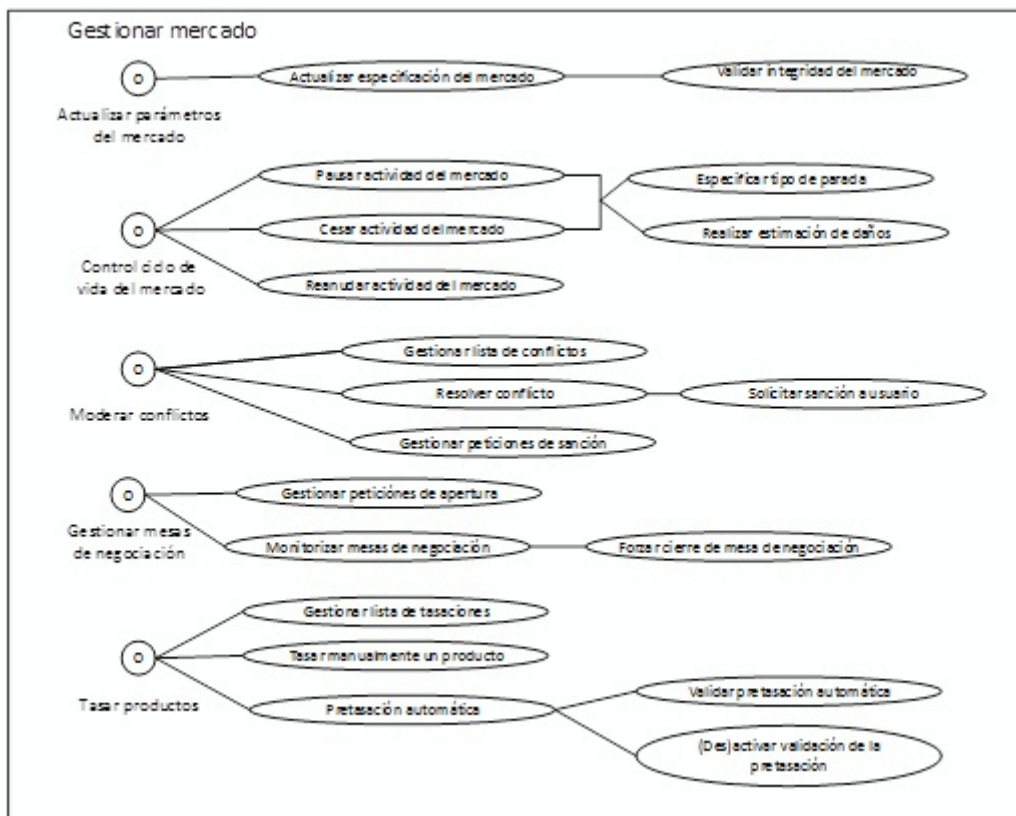


Figura A.3: Diagrama de tareas referente a la administración del mercado



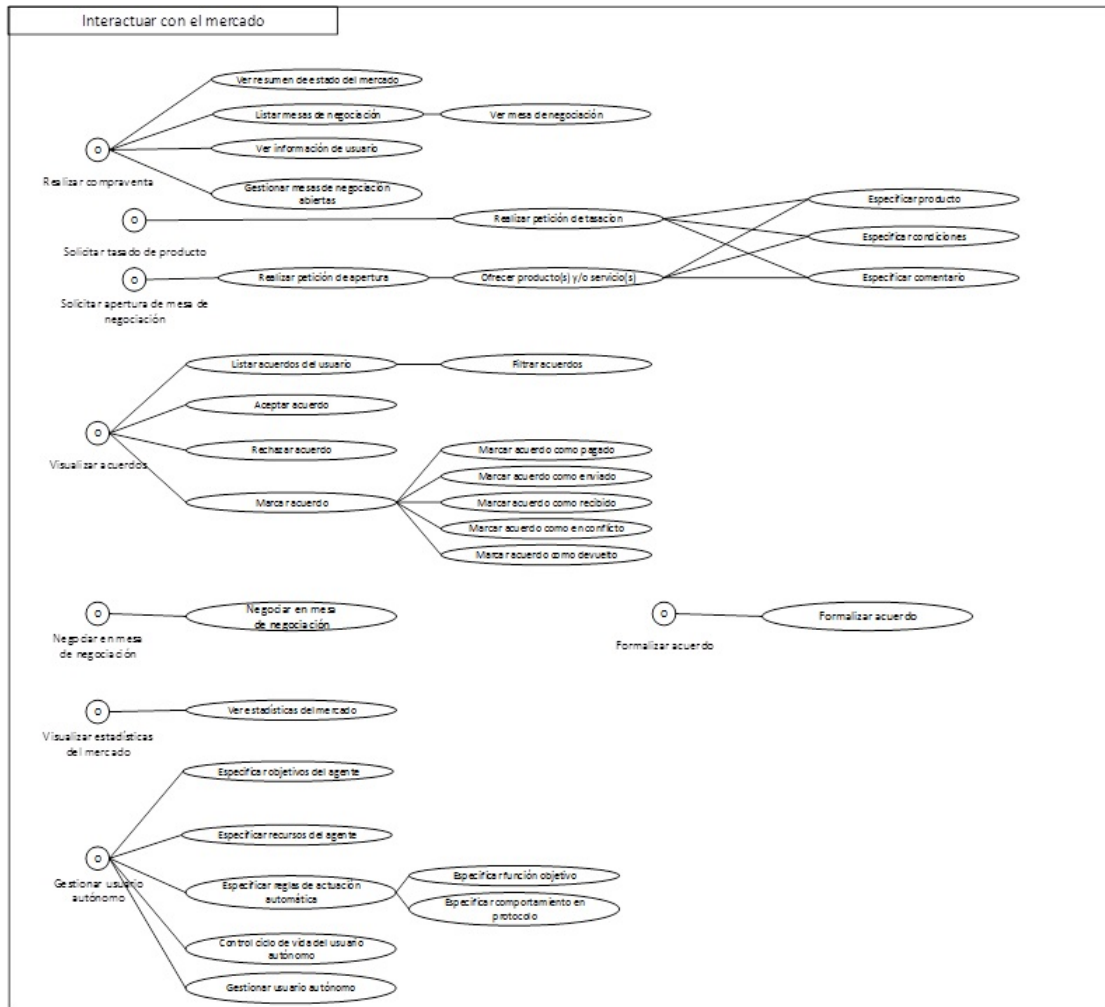


Figura A.4: Diagrama de tareas referente a la interacción con el mercado