



Sistemas informáticos
Curso 2005-2006

HERRAMIENTA DE SIMULACIÓN BURSÁTIL

SOBRE AGENTES SOFTWARE Y WEB SERVICES

Iván Gómez Edo
Itziar Pérez García
Alonso Javier Torres Ortiz

Dirigido por:
Profesor: Manuel Ortega Ortiz de Apodaca
Departamento: Sistemas Informáticos y Programación

Facultad de Informática
Universidad Complutense de Madrid

RESUMEN

El proyecto consiste en la realización de una herramienta para la simulación de un mercado bursátil usando para ello un sistema multiagente. Esta herramienta estará compuesta por un sistema distribuido basado en una arquitectura cliente servidor. El servidor proporcionará las interfaces necesarias para que los agentes y el usuario puedan realizar compra/venta de acciones y recibir información necesaria para su inversión. A su vez los clientes contendrán los sistemas de agentes que realizarán las operaciones que crean convenientes, haciendo fluctuar los precios en bolsa. La interconexión entre el cliente y el servidor se realiza utilizando Web services implementados con la librería Apache Axis sobre el servidor de aplicaciones Apache Tomcat.

SUMMARY

The aim of this project is to develop a tool for stock market simulation with the aid of a multi-agent system. This tool is a distributed system which follows a client-server architecture. The server acts as a simulating environment and provides interfaces so that both clients and agents can buy/sell stocks and receive necessary information to invest in the market. Every client hosts a number of software agents which will perform market operations, thus making the prices fluctuate. The connection between the server and the clients is managed with Web Services implemented using the Apache Axis library and the Apache Tomcat application server.

PALABRAS CLAVE

Bolsa, simulación bursátil, multi-agente, ACL, FIPA, Web service, Axis, WSDL

INDICE

INDICE	5
1. INTRODUCCIÓN	9
1.1 ¿PORQUÉ UNA SIMULACIÓN DE BOLSA?	9
1.2 ¿PORQUÉ UN SISTEMA MULTIAGENTE?	9
1.3 ¿CÓMO INTERACTUARÁN LOS AGENTES CON LA BOLSA?	10
1.4 ¿PORQUÉ WEB SERVICES?.....	10
2. TECNOLOGIAS	11
2.1 JAVA.....	11
2.2 APACHE AXIS	11
2.3 APACHE TOMCAT.....	11
2.4 SISTEMA MULTIAGENTE	11
2.5 XML	12
2.6 JFREECHART	12
2.7 JAVACC.....	12
3. INTRODUCCIÓN A LOS WEB SERVICES.....	12
3.1 ¿QUÉ ES UN WEB SERVICE?.....	12
3.2 VENTAJAS DE LOS WS	13
3.3 INCONVENIENTES DE LOS WS	13
3.4 INTERACCIÓN CLIENTE-SERVIDOR EN LOS WEBSERVICES	13
3.5 FICHERO DE DESCRIPCIÓN DEL WEBSERVICE. PROTOCOLO WSDL	14
3.6 INTERACCIÓN ENTRE WEBSERVICE. PROTOCOLO SOAP.....	15
3.7 UTILIDADES DE AXIS	15
3.7.1 <i>Java2WSDL</i>	16
3.7.2 <i>WSDL2Java</i>	16
3.8 PASOS PARA CREAR UN SERVICIO WEB	16
4.0 CASOS DE USO.....	17
4.1 ÍNDICE DE LOS CASOS DE USO (ORGANIZADOS POR ACTORES).....	17
4.2 CASOS DE USO DEL AGENTE.....	18
4.3 CASOS DE USO DEL USUARIO	23
4.4 CASOS DE USO DEL ADMINISTRADOR	29
4.5 CASOS DE USO DEL SISTEMA DE MENSAJES.....	33
4.6 CASOS DE USO DEL RELOJ	35
4.7 CASOS DE USO DEL SISTEMA DE FLUCTUACIONES	35
4.8 CASOS DE USO DEL SISTEMA DE FLUCTUACIONES	36
5. DESCRIPCIÓN GENERAL DEL SISTEMA	38

5.1 ARQUITECTURA CLIENTE-SERVIDOR.....	38
5.2 PARTES SIGNIFICATIVAS DEL SISTEMA.....	39
5.2.1 <i>Cliente</i>	39
5.2.2 <i>Administrador</i>	39
5.2.3 <i>Servidor</i>	39
5.2.4 <i>Herramienta de agentes</i>	39
5.3 CLIENTE DESDE EL PUNTO DE VISTA DEL SERVIDOR.....	40
5.4 SERVIDOR DESDE EL PUNTO DE VISTA DEL CLIENTE.....	40
5.5 SISTEMA DE AGENTES DENTRO DE LA ARQUITECTURA CLIENTE-SERVIDOR.....	40
5.6 ADMINISTRADOR DEL SISTEMA.....	41
5.7 SISTEMA DE MENSAJES.....	41
5.7.1 <i>Información transmitida por el sistema de mensajes</i>	41
5.7.2 <i>Diagrama de clases de la interacción del cliente con el sistema de mensajes</i>	42
5.8 OBJETOS INTERCAMBIADOS ENTRE EL CLIENTE Y EL SERVIDOR.....	43
5.8.1 <i>Operación</i>	43
5.8.2 <i>Mensajes</i>	43
5.8.3 <i>Información</i>	43
5.9 IMPLEMENTACIÓN DE LA CONEXIÓN CLIENTE-SERVIDOR MEDIANTE WEB SERVICES.....	44
5.9.1 <i>Diagrama de clases de la arquitectura del sistema</i>	44
5.9.2 <i>Fichero WSDL del servidor</i>	45
6. DESCRIPCIÓN DEL SERVIDOR.....	52
6.1 TAREAS DEL SERVIDOR.....	52
6.2 PARÁMETROS DEL SERVIDOR.....	53
6.3 INTERFAZ CON EL ADMINISTRADOR.....	53
6.4 INFORMACIÓN QUE EMITE EL SERVIDOR.....	54
6.4.1 <i>Información en el login</i>	54
6.5 SISTEMA DE EVENTOS.....	54
6.6 ARQUITECTURA DEL SERVIDOR.....	56
6.6.1 <i>ObjetoBolsa</i>	58
6.6.2 <i>Sistema de eventos</i>	59
6.6.3 <i>Gestor de usuarios</i>	60
6.6.4 <i>Interfaz gráfica para el administrador</i>	60
6.6.5 <i>Interfaz para los usuarios</i>	60
7. FLUCTUACIONES DEL SISTEMA.....	62
7.1 ALGORITMO DE SELECCIÓN DEL NUEVO PRECIO.....	63
8. DESCRIPCIÓN DEL CLIENTE.....	64
8.1 ARQUITECTURA DEL CLIENTE.....	65
8.2 INTERACCIÓN DEL CLIENTE CON LA HERRAMIENTA DE AGENTES.....	67

9. SISTEMA MULTIAGENTE	67
9.1 CARACTERÍSTICAS DEL SISTEMA MULTIAGENTE	67
9.2 TIPOS DE AGENTES	68
9.3 ESTRUCTURA GENERAL DE LOS AGENTES	70
9.4 TOMA DE DECISIONES	72
9.5 EJECUCIÓN DE ACCIONES	73
9.6 ESTRUCTURA DEL SISTEMA DE AGENTES.....	74
9.7 INTERACCIÓN AGENTE-BOLSA.....	75
9.8 COMUNICACIÓN ENTRE LOS AGENTES	75
9.9 AGENTES INFORMADORES Y MENSAJES INTERCAMBIADOS DEL PROTOCOLO ACL	77
9.10 POBLACIÓN DINÁMICA DE AGENTES.....	78
9.11 CONFIGURACIÓN DE AGENTES	79
9.12 FICHEROS DE CONFIGURACIÓN.....	79
9.13 PARÁMETROS DEL SISTEMA LOCAL	80
9.14 SISTEMA CLASIFICADOR	81
9.14.1 Codificación de las reglas.....	81
9.14.2 Acciones realizadas por las reglas.....	82
9.14.3 Asignación de beneficio a las reglas	82
9.14.4 Algoritmo genético.....	83
9.14.5 Diagrama de clases del sistema clasificador y el comportamiento.....	84
10. RIESGOS	85
10.1 RIESGOS PERSONALES	85
10.2 RIESGOS TECNOLÓGICOS.....	85
10.3 RIESGOS DE ESTIMACIÓN	86
10.4 SEGUIMIENTO DE LOS RIESGOS	87
10.4.1 Riesgos producidos y planificados:	87
10.4.2 Riesgos producidos y no planificados	88
11. PLANIFICACIÓN	89
11.1 DISTRIBUCIÓN DEL TRABAJO.....	89
11.2 TAREAS REALIZADAS.....	89
11.3 ASIGNACIÓN DE TAREAS.....	91
11.4 DIAGRAMA DE GANTT.....	92
12. PRUEBAS	93
12.1. PRUEBA 1: CÓMO INFLUIR SOBRE UNA EMPRESA	93
12.2. PRUEBA2: INTERACCIÓN ENTRE CLIENTES DE DISTINTAS MÁQUINAS	95
12.3. PRUEBA 3: AVISO DE AMPLIACIÓN DE CAPITAL (DE TIPO SPLIT) SOBRE UN VALOR.....	97
12.4. PRUEBA 4: ACTUACIÓN DE 300 AGENTES Y 2 CLIENTES EN EL SISTEMA.....	98
13. CONCLUSIONES	101

APENDICE A: GLOSARIO.....	102
APENDICE B: MANUAL DE USUARIO.....	105
APENDICE C: MANUAL DE ADMINISTRADOR.....	115
APENDICE D: BIBLIOGRAFÍA.....	126
LIBROS	126
PÁGINAS WEB.....	126
ARTÍCULOS	126

1. INTRODUCCIÓN

1.1 ¿Porqué una simulación de bolsa?

La bolsa es cada vez más un elemento imprescindible para el normal funcionamiento del sistema financiero de una economía capitalista. La función de este mercado financiero es doble. Permite a las empresas obtener financiación externa mediante la emisión de títulos a menor precio del que les supondría el endeudamiento directo con las entidades de crédito. Por otra parte, permite al ofertante de capital (ya sean empresas o particulares) colocar sus sobrantes de liquidez con el objetivo de obtener rentabilidad mediante la revalorización de los títulos o los dividendos repartidos por las sociedades.

Además, la bolsa esta jugando un importante papel como barómetro de la economía, pues el mercado bursátil, con sus oscilaciones, suele utilizarse como indicador de la evolución que esta siguiendo la economía.

Por tanto sería interesante poder tener una modelación de las posibles fluctuaciones del mercado bursátil, tanto de las posibles cotizaciones que alcanzaran los distintos valores como de cuáles son las estrategias más favorables para conseguir la mayor rentabilidad de la inversión.

1.2 ¿Porqué un sistema multiagente?

El modelado basado en agentes permite una buena aproximación para simular este tipo de sistemas y controlar buena parte de la cantidad de elementos que intervienen.

Respecto a otros paradigmas, la simulación basada en agentes permite estudiar cómo emergen comportamientos globales (el del mercado) a partir de otros particulares (los de los agentes).

Además, dadas las características de los agentes, es posible definir cómo éstos evolucionan en un mercado continuo, adaptando sus estrategias, o bien introducir eventos inesperados que pueden determinar nuevas evoluciones del mercado. Los ofertantes de capital serían los “brokers”, que mediante la compra/venta de activos financieros, y siguiendo las reglas de la oferta y la demanda, podrían modelar el mercado bursátil.

Con este objetivo en mente, lo primero que tenemos que saber es en dónde van a estar los agentes, “el mundo” en el que se mueven, lo que llamaremos el entorno de simulación o simplemente entorno.

Además del entorno, necesitamos saber cuales son las interfaces mediante las cuales el agente se comunica con el entorno y la forma en la que los agentes interaccionan entre ellos.

El principal objetivo de estos agentes de software sería:

- Mediante diferentes estrategias, elegir el momento adecuado para invertir y el momento para vender.
- Seleccionar las empresas que tienen unas mejores perspectivas y que, por tanto, permitirán rentabilizar las inversiones.
- Intentar adelantarse a las acciones de otros agentes e intentando disminuir la competencia.
- Comunicarse con otros agentes para saber cuales son las preferencias del mercado actualmente.

1.3 ¿Cómo interactuarán los agentes con la bolsa?

Los agentes estarán distribuidos en diferentes máquinas y se comunicarán con el entorno de una forma centralizada mediante un protocolo de invocación de métodos remotos. De forma que los agentes actuarán como un cliente y el entorno de simulación actuará como servidor. Para esta comunicación vamos a utilizar Web services.

1.4 ¿Porqué Web services?

De acuerdo con el W3C, un Web service es un sistema software diseñado para soportar la interoperabilidad máquina-a-máquina sobre una red. Posee una interfaz que es descrita en un lenguaje procesable por el computador y se comunica mediante mensajes. Esto aporta una serie de ventajas que no soportan otros RPC's para Java (como podría ser RMI o CORBA) para nuestro sistema, como puede ser que la comunicación es totalmente portable a cualquier tipo de implementación de los clientes y la mejor legibilidad de las interfaces al ser realizadas en formatos legibles como el WSDL, dado que es texto plano.

2. TECNOLOGIAS

2.1 Java

Java es un lenguaje de programación orientado a objetos. Hemos empleado este lenguaje para implementar nuestra aplicación porque es multiplataforma y multihilo, además de por la gran cantidad de librerías que existen y que nos permiten contar con cierta funcionalidad ya implementada (como la librería de gráficas JFreeChart).

2.2 Apache Axis

Apache Axis es un “framework” open source de Web services desarrollado sobre Java y XML que consiste en una implementación del servidor SOAP siguiendo las directivas del W3C y varias utilidades y APIS's para generar y desplegar aplicaciones de Web services.

2.3 Apache Tomcat

Apache Tomcat es un contenedor Web desarrollado por la Fundación Apache. Tomcat implementa las especificaciones de Sun Microsystems acerca de servlets y de JavaServer Pages (JSP), proveyendo de un entorno para el código Java para ejecutarse en cooperación con un servidor web. Añade herramientas para la configuración y el mantenimiento pero también puede ser configurado para editar los ficheros de configuración que son normalmente ficheros XML. Tomcat también incluye un servidor web interno y puede ser considerado como un servidor web.

2.4 Sistema Multiagente

Un agente software es una abstracción, un modelo lógico que describe el software que actúa por un usuario o por un programa. Este agente es autónomo y tiene autoridad para decidir cuando (y si) una acción es apropiada. La idea es que los agentes no son estrictamente invocados para una tarea, sino que se activan por ellos mismos.

Un sistema multiagente es un sistema compuesto por varios agentes, que son capaces colectivamente de alcanzar metas que son difíciles de conseguir por agentes individuales o sistemas monolíticos.

2.5 XML

Toda la configuración del sistema se encuentra en ficheros XML, es un lenguaje de marcado de propósito general para crear lenguajes de marcado particulares, en nuestro caso la configuración de los diferentes sistemas de la aplicación.

2.6 JFreeChart

Jfreechart es una librería licenciado bajo licencia LGPL para la creación de diagramas Java y que hace fácil para los desarrolladores mostrar diagramas de alta calidad en las aplicaciones. En una aplicación de bolsa como la nuestra era muy importante poseer una herramienta para poder visualizar bien todos los gráficos relacionados.

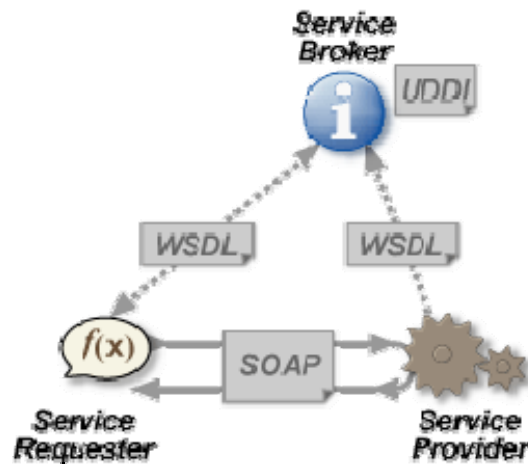
2.7 JavaCC

Es el generador de traductores más popular para usar con aplicaciones Java. JavaCC parte de una especificación de la sintaxis del lenguaje y genera un programa en Java capaz de reconocer frases que concuerden con la sintaxis especificada. Lo hemos empleado para definir las condiciones en el sistema de eventos del servidor.

3. INTRODUCCIÓN A LOS WEB SERVICES

3.1 ¿Qué es un Web Service?

- Es un sistema software que permite la interacción entre máquinas remotas.
- Tiene una interfaz descrita en un lenguaje el cual es interpretable por una máquina (WSDL).
- Otros sistemas interactúan con los Web Services mediante mensajes SOAP (XML), normalmente usando HTTP con una serialización XML en conjunción con otros estándares Web.
- Se pueden publicar la información y la interfaz para ser usada por los WS (UDDI).



3.2 Ventajas de los WS

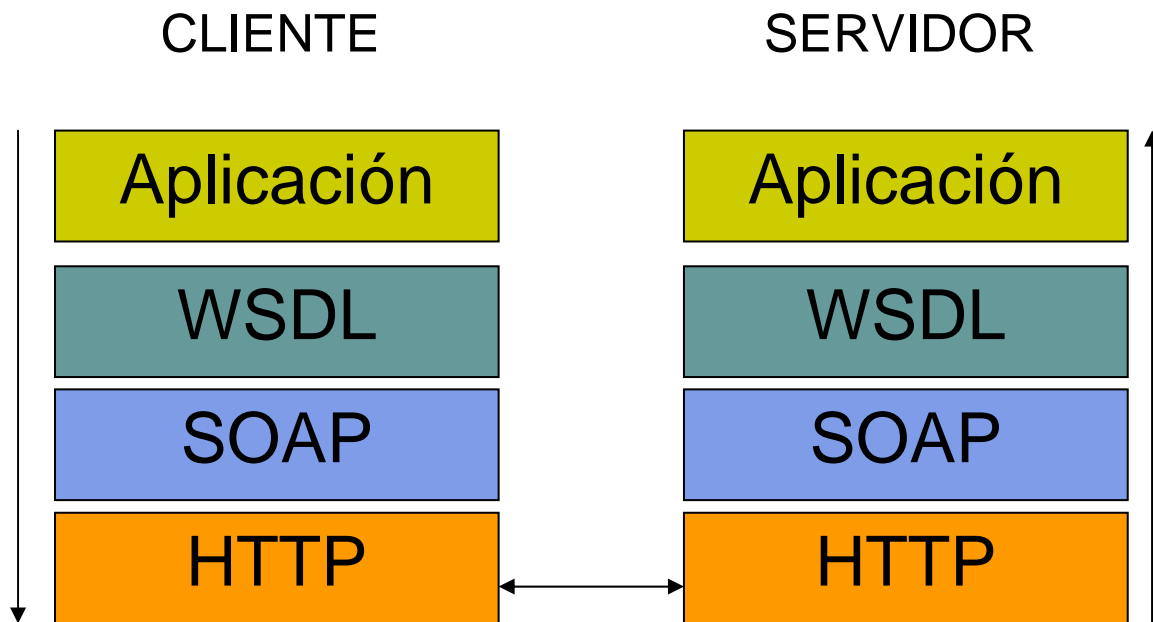
- Permiten interoperabilidad entre distintas aplicaciones en distintas plataformas, independientemente del lenguaje de programación que usen.
- Usan estándares y protocolos abiertos.
- Al utilizar HTTP puede usarse a través de firewalls sin cambiar sus reglas de filtrado. Muchos otros métodos de RPC (RMI, CORBA...) presentan más problemas.
- Permiten la reusabilidad de los distintos componentes.
- Su bajo acoplamiento permiten una aproximación distribuida a la integración de la aplicación.

3.3 Inconvenientes de los WS

- Son menos eficientes que otras formas de RPC.
 - Problema por la decodificación de los mensajes XML.
- La mayoría de las implementaciones actuales no soportan transacciones o las soportan peor que otros RPC's.
 - Este problema se irá solucionando poco a poco.

3.4 Interacción cliente-servidor en los WebServices

Vemos a continuación como interactúan los protocolos que intervienen en la comunicación.



Partimos de la aplicación cliente, que mediante el protocolo WSDL conoce la “interfaz remota” de la aplicación servidor. Mediante el protocolo SOAP transforma su llamada en una llamada HTTP a la aplicación servidor. Esta a su vez realiza el mecanismo contrario, para traducir la llamada a la aplicación servidor.

3.5 Fichero de descripción del WebService. Protocolo WSDL

WSDL (Web Service Description Language). Es un formato XML que describe la interfaz pública del Web Service.

Describe la forma de comunicación:

- Requisitos del protocolo.

- Formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.

- Las operaciones y mensajes que soporta se describen en abstracto.

```

<!-- Definición de la interface -->
<wsdl:portType name="MonitorFilosofos">
  <wsdl:operation name="permisoComer" parameterOrder="nFilosofo">
    <wsdl:input message="impl:permisoComerRequest" name="permisoComerRequest"/>
    <wsdl:output message="impl:permisoComerResponse" name="permisoComerResponse"/>
  </wsdl:operation>

  <wsdl:operation name="finComer" parameterOrder="nFilosofo">
    <wsdl:input message="impl:finComerRequest" name="finComerRequest"/>
    <wsdl:output message="impl:finComerResponse" name="finComerResponse"/>
  </wsdl:operation>

  <wsdl:operation name="dameEstado">
    <wsdl:input message="impl:dameEstadoRequest" name="dameEstadoRequest"/>
    <wsdl:output message="impl:dameEstadoResponse" name="dameEstadoResponse"/>
  </wsdl:operation>
</wsdl:portType>

```

Los servicios son definidos usando 6 elementos principales.

- Tipos de datos intercambiados por el WS.
- Mensajes abstracción de los datos intercambiados.
- Tipos de puerto. Conjunto de operaciones compuesto por mensajes (Normalmente entrada y salida).
- Asociaciones. Define el protocolo concreto y el formato de datos.
- Puerto. Especifica la dirección de una asociación.
- Servicios. Junta los puertos necesarios del servicio.

3.6 Interacción entre WebService. Protocolo SOAP

SOAP (Simple Object Access Protocol). Es un protocolo para intercambiar mensajes basados en XML en una red. Normalmente usando HTTP.

Hay algunos tipos diferentes de mensajes SOAP, el más utilizado es el patrón RPC.

```

<?xml version='1.0' encoding='UTF-8' ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <impl:permisoComerRequest xmlns:impl="pr4">
      <nFilosofo>1</nFilosofo>
    </impl:permisoComerRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

3.7 Utilidades de AXIS

3.7.1 *Java2WSDL*

Proporcionando una interfaz remota genera el fichero *.wsdl con la descripción del servicio Web con esa misma interfaz.

3.7.2 *WSDL2Java*

Dado un fichero *.wsdl genera las clases necesarias para implementar el servicio Web tanto del lado del cliente como del servidor.

Ambas utilidades se pueden usar como “tasks” de ANT.

3.8 Pasos para crear un servicio web

- Crear el archivo .wsdl

Creamos la interfaz remota (igual que en el caso de RMI) y mediante la utilidad Java2WSDL convertimos esta interfaz en un fichero de descripción WSDL.

- Generar las clases que implementen la interfaz definida por el archivo .wsdl
- Una vez creado el fichero de descripción WSDL, usamos la aplicación WSDL2Java para que se nos genere los ficheros Stub y Skelton de la aplicación Servidor. Implementamos el fichero Skelton con la funcionalidad deseada.

- Iniciar el servidor de aplicaciones.

Simplemente iniciamos el Tomcat que descubrirá la instalación del Axis

- Desplegar el Servicio Web en el servidor de aplicaciones.

Una vez instalada la aplicación Axis en el directorio de “webapps” del Tomcat, copiamos las clases necesarias a la carpeta WEB-INF\classes o si esta comprimido en un fichero *.jar a WEB-INF\lib. Después tenemos que usar la aplicación de Axis AdminClient, que sirve para desplegar la aplicación.

- Crear la aplicación cliente que haga uso del servicio.

Implementar dentro de los stubs/skeltons las clases necesarias para la funcionalidad deseada.

4.0 CASOS DE USO

4.1 Índice de los casos de uso (organizados por actores)

- Agente
 - CU-01: Introducir una Operación.
 - CU-02: Cancelar una Operación
 - CU-04: Solicitar información Histórica
 - CU-05: Mandar un mensaje a otro agente
 - CU-06: Recibir un mensaje de un agente
 - CU-07: Salir del sistema
 - CU-08: Hacer una petición de un servicio
- Usuario
 - CU-01: Introducir una Operación.
 - CU-02: Cancelar una Operación
 - CU-03: Solicitar información de una Empresa
 - CU-04: Solicitar información Histórica
 - CU-09: Registrar a un usuario
 - CU-10: Hacer login en el sistema
 - CU-11: Consultar las fluctuaciones de la bolsa
 - CU-12: Hacer uso de herramientas de inversión
 - CU-13: Iniciar la herramienta de agentes
 - CU-14: Parar la herramienta de agentes
 - CU-15: Consultar las acciones que están realizando los agentes
 - CU-16: Listado de la Cartera
- Administrador
 - CU-17: Iniciar la simulación
 - CU-18: Pausar simulación
 - CU-19: Finalizar simulación.
 - CU-20: Comprueba las operaciones en bolsa.
 - CU-21: Comprueba el precio de las acciones.
 - CU-22: Modificar la configuración de las herramientas de agentes externas
 - CU-23: Introducir eventos en el sistema de eventos.
- Sistema de mensajes
 - CU-24: Notificación a los agentes de un evento de bolsa.

- CU-25: Notificación a los agentes de la realización de una operación.
- Reloj
 - CU-26: Ejecución de un “Paso” de tiempo
- Sistema de fluctuaciones
 - CU-27: Generar el nuevo valor de una acción
- Sistema de operaciones
 - CU-28: Cruzar las operaciones.

4.2 Casos de uso del agente.

CU-01: Introducir una Operación	
Objetivo en contexto	Un usuario (ya sea cliente o agente) desea realizar una determinada operación con un título.
Entradas	Nombre del título, tipo de operación, Numero de títulos y precio del título.
Precondiciones	-
Salidas si éxito	Genera una nueva petición.
Salida si fallo	Indica el motivo del fallo.
Postcondición si éxito	Se crea una entrada en la ventana de movimientos y se envía la petición al servidor mediante la herramienta de agentes.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario/Agente, Sistema de operaciones
Secuencia si éxito	P-1.- El usuario introduce los datos correspondientes a una orden. P-2.- El sistema verifica si son correctos y los envía al servidor para ser procesados. P-3.- La orden queda registrada en la ventana.
Secuencia si fallo	A-1.- Aparece la ventana emergente informando del fallo. A-2.- El sistema no realiza nada.
CU-02: Cancelar una Operación	

Objetivo en contexto	Eliminar una orden lanzada que aún no ha sido ejecutada.
Entradas	IdOperación.
Precondiciones	Que exista alguna acción lanzada sin ejecutar.
Salidas si éxito	Se elimina la petición seleccionada.
Salida si fallo	Se avisa con un mensaje emergente.
Postcondición si éxito	Hay una entrada menos en la lista de acciones a ejecutar.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario/Agente, Sistema de Operaciones
Secuencia si éxito	P-1.- El usuario cancela una orden. P-2.- La orden queda eliminada en caso que aún no haya sido procesada.
Secuencia si fallo	A-1: Aparece la ventana emergente informando del fallo. A-2: El sistema no realiza nada.

CU-03: Solicitar información de una empresa	
Objetivo en contexto	Obtener datos relevantes de una empresa seleccionada.
Entradas	Nombre de la empresa especificada.
Precondiciones	Que exista la empresa deseada.
Salidas si éxito	La información de la empresa se visualiza en la ventana correspondiente.
Salida si fallo	Se avisa con un mensaje emergente.
Postcondición si éxito	En la vista del cliente aparece la información relacionada con la empresa.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario
Secuencia si éxito	P-1: El cliente pide información de una determinada empresa. P-2: El sistema comprueba que esa información esta disponible y se lo entrega.

Secuencia si fallo	A-1: Aparece la ventana emergente informando del fallo. A-2: El sistema no realiza nada.
---------------------------	---

CU-04: Solicitar información Histórica	
Objetivo en contexto	Obtener datos Históricos sobre una empresa seleccionada en un determinado espacio temporal.
Entradas	Nombre de la empresa especificada, fecha de inicio y fecha de fin.
Precondiciones	Que exista la empresa deseada.
Salidas si éxito	Usuario: En un nuevo gráfico aparece el histórico de la empresa deseada. Agente: Recibe la información histórica.
Salida si fallo	Se avisa con un mensaje emergente.
Postcondición si éxito	Usuario: En la ventana de información de empresa aparece el gráfico con su histórico Agente: El agente recibe una estructura de datos con la información histórica
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario/Agente.
Secuencia si éxito	P-1: El usuario pide el histórico de una determinada empresa. P-2: El sistema comprueba que esa información esta disponible y se lo entrega.
Secuencia si fallo	A-1: Aparece la ventana emergente informando del fallo. A-2: El sistema no realiza nada.

CU-05: Mandar un mensaje a otro agente	
Objetivo en contexto	Mandar información a otro agente en el sistema.
Entradas	Nombre del agente destino, mensaje que desea enviar.

Precondiciones	Que el agente esté dado de alta en el sistema de mensajes. El destinatario tiene que existir.
Salidas si éxito	-
Salida si fallo	Excepción por no existencia del agente.
Postcondición si éxito	El buzón del agente destino posee un nuevo mensaje.
Postcondición si fallo	El sistema no se modifica.
Actores	Agente
Secuencia si éxito	P-1: El agente crea un objeto de mensaje y rellena los campos necesarios. P-2: Indica el destinatario del mensaje. P-3: El agente manda el mensaje a través de su propio buzón de mensajes. P-4: El agente destinatario recibe el mensaje.
Secuencia si fallo	A-1: El agente destino no existe. A-2: Se le comunica al agente emisor que ha habido un problema con su envío.

CU-06: Recibir un mensaje de un agente	
Objetivo en contexto	Recibir un mensaje para actuar en consecuencia.
Entradas	Una plantilla que encaja con el tipo de mensajes que desea recibir el agente.
Precondiciones	El agente ha sido dado de alta en el sistema de mensajes.
Salidas si éxito	-
Salida si fallo	Error en la recepción del mensaje
Postcondición si éxito	El agente invoca el método asociado a la invocación del mensaje.
Postcondición si fallo	No hay cambios en el sistema.
Actores	Agentes.
Secuencia si éxito	P-1: El agente crea un objeto de plantilla que

	<p>encaja con el mensaje que desea recibir.</p> <p>P-2. El agente inserta la plantilla junto con los métodos que desea invocar cuando se reciba el mensaje y el número de mensajes que desea.</p> <p>P-3: Eventualmente se recibirá un mensaje que encajará con la plantilla e invocará los métodos definidos en la recepción del mensaje.</p>
Secuencia si fallo	A-1: Al agente le comunican el error.

CU-07: Salir del sistema	
Objetivo en contexto	El agente abandona la simulación.
Entradas	-
Precondiciones	El agente debería estar previamente dado de alta en la simulación.
Salidas si éxito	Se informa al agente de que ha salido de la simulación con éxito.
Salida si fallo	El único posible fallo es que el agente no estuviera en previamente en la simulación por tanto no se le informa.
Postcondición si éxito	El agente sale de la simulación.
Postcondición si fallo	-
Actores	Agente
Secuencia si éxito	<p>P-1: El agente solicita la salida de la simulación al sistema.</p> <p>P-2: El sistema informa al agente de que ya le ha dado de baja.</p> <p>P-3: El agente devuelve los recursos que tuviera al sistema.</p>
Secuencia si fallo	-

CU-08: Hacer una petición de un servicio	
Objetivo en contexto	Petición de un servicio
Entradas	El servicio que desea el agente.

Precondiciones	El agente esté dado de alta en el sistema de mensajes.
Salidas si éxito	Lista con los identificadores de los mensajes que pueden proporcionar el servicio.
Salida si fallo	Una lista vacía.
Postcondición si éxito	El sistema no se modifica.
Postcondición si fallo	El sistema no se modifica.
Actores	Agente.
Secuencia si éxito	<p>P-1: El agente solicita el servicio.</p> <p>P-2: Se le devuelve una lista con los identificadores de los agentes que pueden proporcionar el servicio.</p> <p>P-3: El agente decide si pedirle el servicio al agente determinado o no.</p> <p>P-4: Si decirle pedirle el servicio tiene que enviar un mensaje de REQUEST al agente.</p>
Secuencia si fallo	P-1: Si no hay agentes que proporcionan el servicio, o hay un error en la comunicación se devuelve al agente unan lista vacía.

4.3 Casos de uso del usuario

CU-09: Registrar a un usuario	
Objetivo en contexto	Darse de alta en el sistema para comenzar a operar.
Entradas	Nombre de usuario y contraseña.
Precondiciones	-
Salidas si éxito	Comienza a ejecutar el programa.
Salida si fallo	Indica el fallo y vuelve a pedir el nombre
Postcondición si éxito	Nuevo usuario en la base de datos del sistema asignándole un identificador, se crean los agentes asociados al usuario.
Postcondición si fallo	El sistema no se modifica.

Actores	Agente y BBDD del servidor
Secuencia si éxito	P-1: Un nuevo cliente dispone del programa e introduce su nombre por primera vez. P-2: El sistema registra dicha información y le da permiso para empezar.
Secuencia si fallo	A-1: Aparece la ventana emergente informando del fallo. A-2: El sistema no realiza nada.

CU-10: Hacer login en el sistema	
Objetivo en contexto	El cliente entra en el sistema para incorporarse a la simulación.
Entradas	Nombre de usuario y contraseña
Precondiciones	Que el cliente se haya registrado previamente en el sistema
Salida sí éxito	-
Salida si fallo	Se muestra al usuario un mensaje de error.
Postcondición si éxito	Se incorpora al usuario (y a los agentes que se simulan en su máquina) a la simulación. Su información de empresas y de acciones se actualiza.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario, sistema de mensajes, base de datos, herramienta de agentes.
Secuencia si éxito	P-1: El usuario introduce su nombre de usuario y contraseña. Si no hay ninguna simulación activa, ir a A-1 P-2: Se consulta en la base de datos si el usuario existe previamente. Si no, ir a A-2. P-3: Si el usuario existe, se le introduce en la simulación, y actualiza su información de acciones y empresas. Asimismo se crean distintos tipos de agentes según los parámetros

	iniciales de simulación.
Secuencia si fallo	<p>A-1: No se ha podido conectar por no haber ninguna simulación activa. Se le notifica al usuario mediante un mensaje de error.</p> <p>A-2: No existe ningún usuario con ese nombre y contraseña. El mensaje indicará al usuario que se registre o que revise los datos introducidos.</p>
CU-11: Consultar las fluctuaciones de la bolsa	
Objetivo en contexto	El cliente está interesado en ver, por medio de charts, la evolución de una acción o un índice en la jornada bursátil del día.
Entradas	Seleccionar el valor o índice a visualizar
Precondiciones	Que el cliente esté dentro de la simulación
Salida si éxito	Chart de la acción o índice
Salida si fallo	Se muestra al usuario un mensaje de error.
Postcondición si éxito	El sistema no se modifica.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario, sistema de mensajes, sistema de fluctuaciones.
Secuencia si éxito	<p>P-1: El usuario elige de un scroll el valor o índice que está interesado en visualizar.</p> <p>P-2: Accede a los datos históricos que tiene guardados en su máquina (los recibe cuando se registra y se actualizan periódicamente).</p> <p>P-3: Se dibuja el chart junto con información de la empresa (volumen transaccionado, precio actual de la acción)</p> <p>P-4: Se repiten los pasos 2 y 3 cada vez que se recibe una actualización de dicho valor.</p>
Secuencia si fallo	A-1: Si en algún momento se produce un error de conexión con el servidor, se muestra al usuario un mensaje de error.

CU-12: Hacer uso de herramientas de inversión	
Objetivo en contexto	El cliente emplea herramientas bursátiles (Estocástico y Volumen) para intentar averiguar las tendencias del mercado.
Entradas	-
Precondiciones	Que el cliente esté dentro de la simulación.
Salida si éxito	Devuelve un nuevo gráfico en el cual nos determina el resultado de la herramienta utilizada.
Salida si fallo	Se muestra al usuario un mensaje de error.
Postcondición si éxito	El sistema no se modifica.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario
Secuencia normal	<p>P-1: El usuario selecciona la herramienta.</p> <p>P-2: Se introducen los parámetros necesarios para realizar la predicción fecha inicial y fecha final del periodo que desea consultar.</p> <p>P-3: Se muestra al usuario el RESULTADO DE ESTE TIPO DE PROGRAMAS una grafica que muestra los valores.</p>
Secuencia alternativa	A-1: Si se introduce algún parámetro erróneo o sale algo mal, se muestra al usuario un mensaje de error.

CU-13: Iniciar herramienta de agentes	
Objetivo en contexto	Poner en funcionamiento el contenedor local de agentes del usuario.
Entradas	<ul style="list-style-type: none"> - Número de agentes del sistema. - Número de ciclos que los agentes esperan entre tomas de decisiones de media. - Frecuencia de ciclo del contenedor de acciones.
Precondiciones	El usuario debe estar registrado y dado de alta

	en el sistema.
Salidas si éxito	Se observa por interfaz gráfica cuales son los agentes creados y cuales son sus interacciones con el entorno.
Salida si fallo	Se informa del error al usuario.
Postcondición si éxito	El sistema de agentes se inicia y los agentes empiezan a interactuar.
Postcondición si fallo	El sistema no se modifica.
Actores	Agentes, Usuario, Sistema multiagente
Secuencia si éxito	P-1: El usuario presiona el botón de iniciar la herramienta de agentes. P-2: Sale un dialogo para poder introducir los parámetros. P-3: El sistema comienza.
Secuencia si fallo	A-1: Se muestra al usuario un mensaje informando del error.

CU-14: Parar herramienta de agentes	
Objetivo en contexto	Detener temporalmente la herramienta de agentes.
Entradas	-
Precondiciones	La herramienta debe estar iniciada y el usuario dado de alta en el sistema.
Salidas si éxito	Los agentes detienen sus interacciones con el sistema.
Salida si fallo	-
Postcondición si éxito	Los agentes detienen sus interacciones con el sistema.
Postcondición si fallo	No hay cambios en el sistema.
Actores	Usuario, Agentes, Sistema Multiagentes
Secuencia si éxito	P-1: El usuario presiona el botón para parar la herramienta de agentes. P-2: Los agentes detienen sus interacciones.

Secuencia si fallo	A-1: Se informa al usuario del error.
---------------------------	---------------------------------------

CU-15: Consultar las acciones que están realizando los agentes	
Objetivo en contexto	Poder visualizar las operaciones que realizan los agentes y las notificaciones que les llegan desde la bolsa.
Entradas	-
Precondiciones	El sistema de agentes debe estar iniciado.
Salidas si éxito	Las operaciones y las notificaciones de los agentes del sistema.
Salida si fallo	Mensaje de error.
Postcondición si éxito	El sistema no se modifica
Postcondición si fallo	El sistema no se modifica.
Actores	Agentes, Usuario, sistema de agentes.
Secuencia si éxito	P-1: El usuario inicia la herramienta de agentes. P-2: Se muestra por interfaz gráfica las interacciones de los agentes.
Secuencia si fallo	A-1: Se informa del error.

CU-16: Listado de la Cartera	
Objetivo en contexto	Ver un listado de todas las acciones que dispone un usuario en su cartera de valores.
Entradas	-
Precondiciones	Disponer de acciones en la cartera.
Salidas si éxito	Nos muestra en un lugar de la pantalla el listado del contenido de la cartera.
Salida si fallo	-
Postcondición si éxito	Aparece la nueva gráfica debajo de la gráfica de intradía que muestra los resultados obtenidos.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario del servidor.
Secuencia si éxito	P-1: Pincha en el menú despegable de la gráfica y selecciona la herramienta deseada.

	P-2: Introducir el periodo a investigar. P-3: Obtenemos la gráfica correspondiente
Secuencia si fallo	-

4.4 Casos de uso del administrador

CU-17: Iniciar la simulación	
Objetivo en contexto	El administrador arranca el servidor para iniciar una simulación
Entradas	<p>Parámetros de la simulación:</p> <ol style="list-style-type: none"> 1. Velocidad. 2. <u>Tiempo de cambio</u>: Cada cuanto tiempo hay que recalcular el valor de las acciones. 3. <u>Interrumpido</u>: Indica si queremos que la simulación se interrumpa al llegar al final de la jornada 4. Tiempo de preapertura 5. Tiempo de cierre 6. <u>Configuración inicial de los agentes</u>: Cada usuario que se conecte generará un número de agentes según las capacidades de su máquina. Esto determinará que porcentaje de agentes habrá de cada tipo.
Precondiciones	Que no haya ninguna simulación en proceso
Salida si éxito	-
Salida si fallo	Se muestra al administrador un mensaje de error.
Postcondición si éxito	Comienza la simulación como una nueva jornada bursátil. El precio inicial de los valores será el de la simulación anterior.
Postcondición si fallo	-

Actores	Administrador, herramienta de agentes, base de datos.
Secuencia si éxito	P-1: El administrador introduce los parámetros de inicio de la bolsa (véase Entradas) P-2: La simulación se inicia con el último valor del histórico.
Secuencia si fallo	A-1: Si falta algún parámetro se informa al administrador.

CU-18: Pausar simulación	
Objetivo en contexto	Pausar la simulación para mejor control de la misma.
Entradas	-
Precondiciones	El sistema tiene que estar iniciado previamente.
Salidas si éxito	-
Salida si fallo	Mensaje de error al usuario.
Postcondición si éxito	El sistema se detiene
Postcondición si fallo	Se sale del sistema.
Actores	Administrador
Secuencia si éxito	P-1: El administrador pulsa el botón de pausar la simulación. P-2: La simulación se detiene
Secuencia si fallo	A-1: Mensaje de error

CU-19: Finalizar simulación	
Objetivo en contexto	Poner fin a la simulación bursátil.
Entradas	-
Precondiciones	El sistema tiene que estar iniciado previamente.
Salidas si éxito	-
Salida si fallo	Mensaje de error al usuario.
Postcondición si éxito	El sistema finaliza
Postcondición si fallo	Se sale del sistema.
Actores	Administrador

Secuencia si éxito	P-1: El administrador pulsa el botón de finalizarla simulación. P-2: La simulación finaliza
Secuencia si fallo	A-1: Mensaje de error

CU-20: Comprueba las operaciones en bolsa	
Objetivo en contexto	Observar cuales han sido las operaciones efectuadas en la bolsa simulada.
Entradas	-
Precondiciones	El sistema debe estar iniciado y debe haber operaciones en bolsa para observar.
Salidas si éxito	Las operaciones que deseamos observar.
Salida si fallo	Mensaje de error al usuario.
Postcondición si éxito	El sistema no se modifica.
Postcondición si fallo	El sistema no se modifica.
Actores	Administrador
Secuencia si éxito	P-1: El sistema está iniciado. P-2: Se muestran las operaciones que operan en la bolsa.
Secuencia si fallo	A-1: Mensaje de error.

CU-21: Comprueba precio de las acciones	
Objetivo en contexto	Poder comprobar el precio de las acciones de la bolsa.
Entradas	-
Precondiciones	El sistema debe estar iniciado y la información de las empresas debe ser cargada correctamente.
Salidas si éxito	El precio de las acciones que están en la bolsa.
Salida si fallo	Mensaje de error al usuario.
Postcondición si éxito	El sistema no se modifica.
Postcondición si fallo	El sistema no se modifica.
Actores	Administrador

Secuencia si éxito	P-1: El sistema está iniciado. P-2: Se muestran las operaciones que operan en la bolsa.
Secuencia si fallo	A-1: Mensaje de error.

CU-22: Modificar la configuración de las herramientas de agentes externas	
Objetivo en contexto	El administrador modifica los porcentajes de los tipos de agentes.
Entradas	Los porcentajes de cada tipo de agente. Tienen que sumar 100.
Precondiciones	Que no haya ninguna simulación en marcha.
Salida si éxito	-
Salida si fallo	Se muestra al administrador un mensaje de error.
Postcondición si éxito	Los clientes que se conecten a la simulación crearán sus agentes según la nueva configuración.
Postcondición si fallo	El sistema no se modifica.
Actores	Administrador, herramienta de agentes, sistema de mensajes.
Secuencia si éxito	P-1: El administrador introduce los nuevos porcentajes. Si no suman 100 ir a A-1. P-2: La simulación se iniciará con esta nueva configuración.
Secuencia si fallo	A-1: Si no suman 100 se informará al administrador mediante un mensaje.

CU-23: Introducir eventos en el sistema de eventos.	
Objetivo en contexto	Introducir un nuevo evento en el sistema de eventos.
Entradas	Las características del evento.
Precondiciones	-

Salidas si éxito	-
Salidas si fallo	Información del motivo por el que se produce el error.
Postcondición si éxito	El evento queda insertado en el sistema de eventos.
Postcondición si fallo	El sistema no se modifica.
Actores	Usuario Web, sistema
Secuencia si éxito	P-1: El administrador pulsa el botón correspondiente a la inserción de un evento. P-2: Salta una ventana para la introducción de los parámetros del evento. P-3: Una vez el administrador introduce los parámetros del evento, pulsa aceptar y el evento se introduce en el sistema de eventos.
Secuencia si fallo	A-1: Error en la introducción de los parámetros. Se dirá al administrador los errores para que pueda corregirlos.

4.5 Casos de uso del sistema de mensajes.

CU-24: Notificación a los agentes de un evento de bolsa.	
Objetivo en contexto	Comunicar un evento a los agentes dados de alta.
Entradas	-
Precondiciones	Que se produzcan las condiciones para que se ejecute el evento.
Salidas si éxito	-
Salidas si fallo	Información del motivo por el que se produce el error.
Postcondición si éxito	El evento queda insertado en el sistema de eventos.
Postcondición si fallo	El sistema no se modifica.

Actores	Sistema de mensajes, Sistema de eventos
Secuencia si éxito	<p>P-1: El administrador pulsa el botón correspondiente a la inserción de un evento.</p> <p>P-2: Salta una ventana para la introducción de los parámetros del evento.</p> <p>P-3: Una vez el administrador introduce los parámetros del evento, pulsa aceptar y el evento se introduce en el sistema de eventos.</p>
Secuencia si fallo	<p>A-1: Error por mala introducción de los datos. Se vuelve a pedir al administrador que introduzca los datos.</p>

CU-25: Notificación a los agentes de la realización de una operación.	
Objetivo en contexto	Comunicar al agente que introdujo la operación que se ha producido la compra o venta.
Entradas	-
Precondiciones	Que el agente correspondiente haya introducido una operación y esta se haya cruzado con otra de signo contrario con éxito.
Salidas si éxito	-
Salidas si fallo	-
Postcondición si éxito	El agente queda notificado de que se ha realizado su operación.
Postcondición si fallo	El sistema no se modifica.
Actores	Agente, Sistema de mensajes, Sistema de fluctuaciones
Secuencia si éxito	<p>P-1: El sistema de compra/venta eventualmente cruzará una operación y se realizará una transacción.</p> <p>P-2: El sistema cuando cruce la operación, notifica al sistema de mensajes.</p> <p>P-3: El sistema de mensajes a su vez notifica la</p>

	realización de la operación al agente correspondiente.
Secuencia si fallo	A-1: En caso de que el agente al que se tiene que notificar la operación no este presente se guarda el mensaje a la espera de que vuelva a conectar. A-2: Cuando conecta el agente, se le notifica.

4.6 Casos de uso del reloj

CU-26: Ejecución de un “Paso” de tiempo	
Objetivo en contexto	Comunicar el paso de un paso de reloj para la sincronización.
Entradas	-
Precondiciones	Que haya pasado un tiempo T definido por el administrador desde la última ejecución de un paso.
Salidas si éxito	-
Salidas si fallo	-
Postcondición si éxito	Todos los oyentes del sistema quedan notificados por la producción de un paso.
Postcondición si fallo	El sistema no se modifica.
Actores	Reloj del programa.
Secuencia normal	P-1: Comprobamos que el tiempo del último paso es superior al T definido para la simulación. P-2: Se produce el paso
Secuencia alternativa	A-1: No ha pasado el tiempo T desde el último paso, el sistema sigue esperando para ejecutar.

4.7 Casos de uso del sistema de fluctuaciones

CU-27: Generar el nuevo valor de una acción
--

Objetivo en contexto	Calcular el nuevo valor de la acción en función de la oferta y la demanda.
Entradas	Operaciones sobre una acción.
Precondiciones	Que se haya producido un paso de tiempo.
Salidas si éxito	-
Salidas si fallo	-
Postcondición si éxito	El precio de la acción toma un nuevo valor.
Postcondición si fallo	El sistema no se modifica.
Actores	Sistema de fluctuaciones, Sistema de operaciones, sistema de mensajes.
Secuencia si éxito	<p>P-1: El sistema de operaciones posee en un determinado momento todas las operaciones efectuadas sobre una acción.</p> <p>P-2: Eventualmente se produce un paso de tiempo en el sistema.</p> <p>P-3: El sistema de fluctuaciones solicita al de operaciones la información necesaria para el cálculo del nuevo valor.</p> <p>P-4: Si el precio de la acción cambia, se informa al sistema de mensajes para que lo notifique a los agentes.</p>
Secuencia si fallo	A-1: Si el precio de la acción no cambia el sistema queda como estaba hasta el próximo paso de tiempo.

4.8 Casos de uso del sistema de fluctuaciones

CU-28: Cruzar las operaciones.	
Objetivo en contexto	Cruce de las operaciones de compra y venta introducidas en el sistema.
Entradas	-
Precondiciones	Operaciones existentes en el sistema de

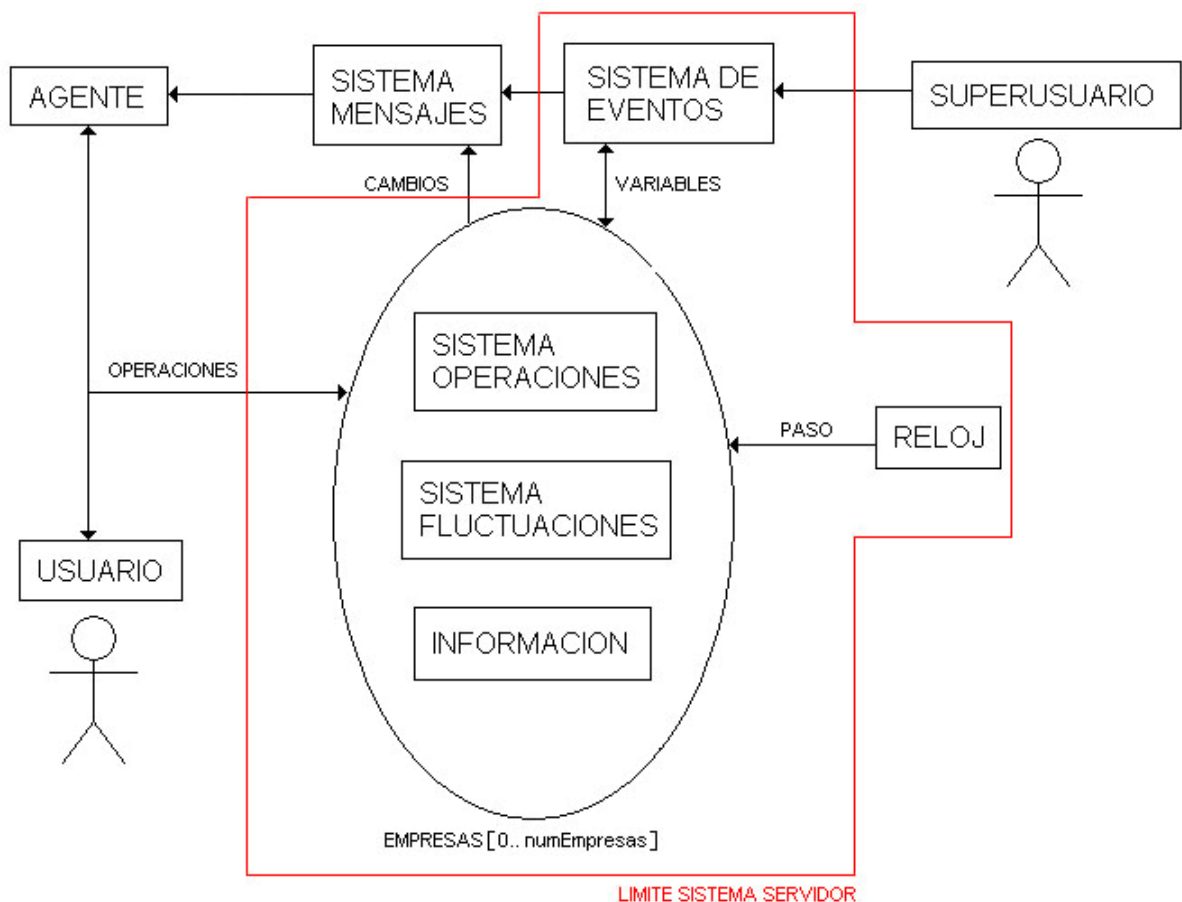
	operaciones.
Salidas si éxito	-
Salidas si fallo	Información del motivo por el que se produce el error.
Postcondición si éxito	Los agentes de compra y de venta se intercambian el dinero correspondiente al valor de la operación.
Postcondición si fallo	El sistema no se modifica.
Actores	Sistema de operaciones, Sistema de mensajes
Secuencia si éxito	<p>P-1: Eventualmente se produce un paso de reloj.</p> <p>P-2: Se intentan cruzar el mayor número de operaciones posibles.</p> <p>P-3: Cuando se consigue cruzar una orden de compra con otra de venta el sistema de operaciones lo comunica al sistema de mensajes.</p>
Secuencia si fallo	A-1: Si al finalizar una jornada en la bolsa no se ha conseguido cruzar la operación cancela automáticamente la operación informando al agente de ello.

5. DESCRIPCIÓN GENERAL DEL SISTEMA

5.1 Arquitectura cliente-servidor

Al comenzar el planteamiento del proyecto, se nos presentaron las opciones de realizar un sistema centralizado o distribuido. Optamos por realizar el sistema de forma centralizada, mediante una arquitectura cliente-servidor. Esto es debido sobre todo a restricciones de la tecnología. Al realizar el sistema mediante WebServices, no podíamos realizar de forma distribuida la ejecución pues tendríamos que tener un servidor de aplicaciones (en nuestro caso Tomcat) por cada uno de los usuarios de la aplicación. Nos pareció mucho más correcto que los clientes hicieran peticiones al servidor que es el que implementa el Webservice.

El servidor a su vez define unas interfaces gráficas para que el administrador del sistema pueda gestionar en todo momento el programa, y poder observar como se esta comportando la simulación.



5.2 Partes significativas del sistema

5.2.1 Cliente

Vamos a considerar que la aplicación cliente va a tener principalmente dos funcionalidades. La primera es proporcionar al usuario de nuestro sistema de las interacciones de entrada/salida necesarias para poder interactuar con el sistema y para poder comprobar los datos de la simulación. La segunda será, mediante la configuración definida por el servidor, alojar agentes que ayuden en la simulación global mediante la herramienta de agentes.

5.2.2 Administrador

El administrador siempre se va a ejecutar en la máquina local en la que se ejecute el servidor. Éste proporciona al administrador una interfaz gráfica, que le permite controlar la simulación en curso y modificar los parámetros de los agentes conectados al servidor, tanto en número de agentes como en estrategias que usan.

5.2.3 Servidor

Se encarga de las funciones correspondientes al entorno, también se encarga del sistema de mensajería, y guarda el histórico, la información de las empresas y las fluctuaciones de la sesión, así como la configuración remota de agentes.

5.2.4 Herramienta de agentes

Para facilitar la escalabilidad vamos a considerar una tercera herramienta que es básicamente una encapsulación de agentes, así tanto el cliente como el servidor podrán tener agentes que estén interactuando con el entorno. La configuración de estas encapsulaciones de agentes se guarda en el servidor, transfiriéndose cuando un nuevo usuario se conecta, el administrador es el único que puede controlar el comportamiento de los agentes y el número de los mismos.

5.3 Cliente desde el punto de vista del servidor

El cliente solo puede ver del servidor su interfaz pública, esto es:

- Alta en el sistema. Login y registro en la simulación.
- Operaciones de bolsa. introducir operaciones, cancelar operaciones.
- Mensajes del sistema. Recepción y envío de mensajes.
- Información de la simulación. Información de las empresas, sus cotizaciones y otros datos.

Cualquier usuario puede darse de alta en el sistema mediante la opción de registro. Esto es requerido sobre todo para que el sistema pueda reconocer unívocamente a un usuario determinado y poder proporcionarle las opciones correspondientes.

Las operaciones de bolsa son compra y venta de acciones. Para realizar una operación es necesario saber la empresa, el precio y la cantidad de acciones que se desean comprar.

En determinados momentos de la simulación, el servidor emitirá avisos al cliente, estos avisos se transmiten del cliente al servidor mediante el sistema de mensajes que también forma parte de la interfaz del servidor. Por motivos de implementación hemos separado las interfaces del sistema de mensajería del resto de funcionalidad en dos WebServices.

En cuanto a la información bursátil, es imprescindible en un sistema de nuestras características proporcionar información de cómo está evolucionando la bolsa pues es la forma en la que tanto los agentes como los clientes poseerán la información necesaria para tomar sus decisiones.

5.4 Servidor desde el punto de vista del cliente

El servidor solo posee la información de los usuarios que están conectados en el sistema. Para que el servidor pueda realizar las notificaciones oportunas, y permita al cliente realizar las operaciones, es necesario que el cliente este registrado y dado de alta.

5.5 Sistema de agentes dentro de la arquitectura cliente-servidor

El cliente tiene funcionalidad de contenedor para el sistema de agentes. Un número determinado de agentes pueden estar en el contenedor que posee el

cliente. Las características y los tipos de agentes los determina el administrador.

5.6 Administrador del sistema

El administrador del sistema tiene principalmente las siguientes funciones:

- Configurar los parámetros de la simulación.
- Observar como funciona la simulación y poder controlar el inicio y finalización de la misma.
- Configurar el comportamiento general de los agentes, mediante los ficheros de configuración.
- Controlar el sistema de eventos (introducir nuevos, o borrar/activar los ya insertados)
- Controlar el acceso de los usuarios, y poder ver los usuarios registrados y conectados.
- Poder monitorizar todas las operaciones que se manejan en el sistema.

5.7 Sistema de mensajes

El sistema de mensajes es una de las formas en las que se pueden comunicar el cliente y el servidor. Es un sistema de comunicación unidireccional que no asegura la entrega del mensaje.

Consiste en dos partes, un emisor y un receptor. El emisor implementa una interfaz remota usando para la implementación un servicio Web.

En cuanto al receptor, es un hilo el cuál se bloquea de forma remota en caso de que no existan mensajes en el servidor.

En la parte del cliente, se usa un manejador de mensajes para distribuir los mensajes donde corresponda dentro del cliente.

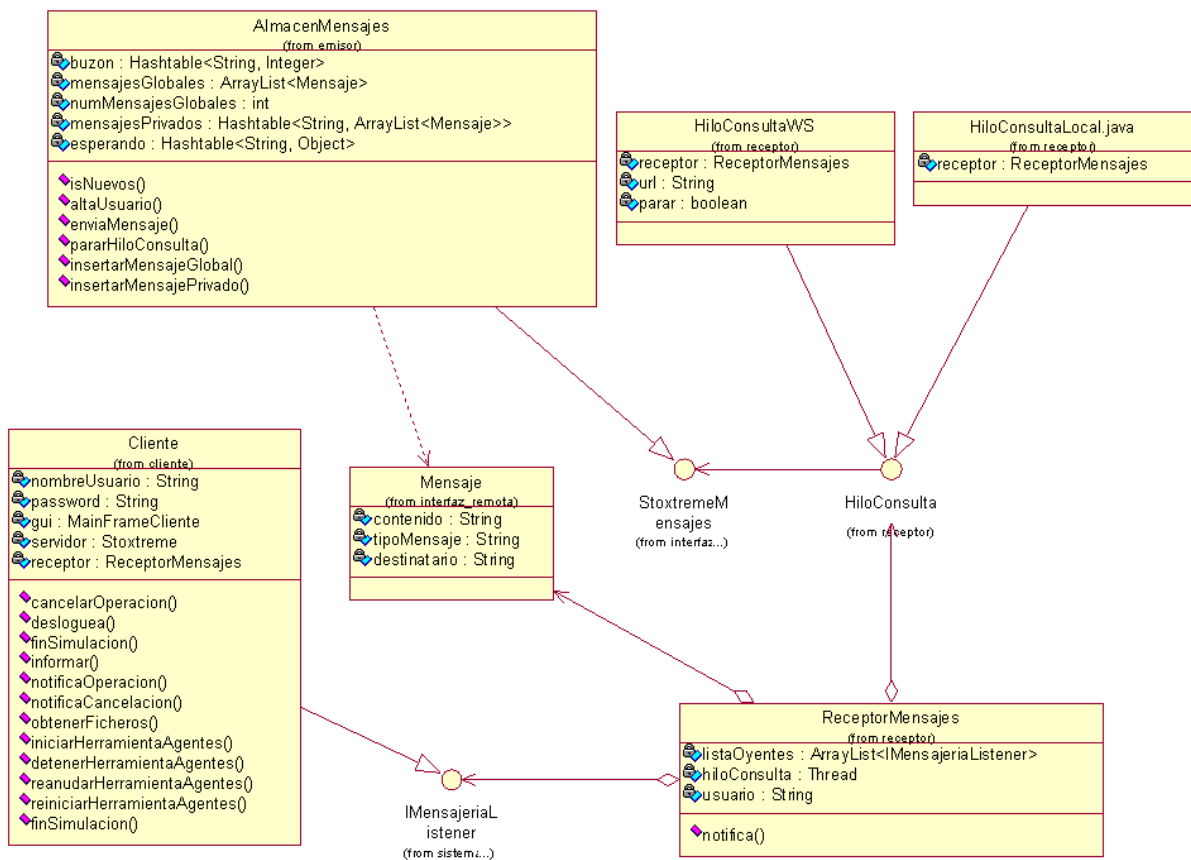
5.7.1 Información transmitida por el sistema de mensajes.

- Actualización de precios. Cada vez que cambie el precio de una acción, el servidor envía el nuevo precio a los usuarios para que se actualice la información de que dispone.

- Notificación de la transacción. Cuando una operación de compra o venta se lleva a cabo con éxito, el servidor avisa al cliente o clientes correspondientes.
- Notificación de los eventos. El servidor avisa al usuario de cualquier evento lanzado por el administrador, para que pueda invertir con conocimiento de causa.

5.7.2 Diagrama de clases de la interacción del cliente con el sistema de mensajes.

Observamos en el siguiente mensaje la estructura del sistema de mensajes. Y su interacción con el cliente. El cliente implementa la interfaz de recepción de mensajes `IMensajeriaListener`, y el Receptor de mensajes enviará el mensaje cuando le llegue. El Hilo de consulta se queda esperando en el método remoto a la recepción de un mensaje e informa al receptor de mensajes del nuevo mensaje.



5.8 Objetos intercambiados entre el cliente y el servidor

Básicamente existen tres tipos de objetos intercambiados entre el cliente y el servidor.

5.8.1 Operación

Es el tipo básico sobre el que actúa la bolsa. Encapsula la información necesaria a la hora de realizar una compra o una venta:

- Identificador del agente que ha realizado la operación.
- Identificador único de la operación.
- Empresa sobre la que se realiza la operación.
- Número de acciones que desea intercambiar.
- Precio que máximo que esta dispuesto a pagar (en caso de ser una compra) o precio al mínimo que desea (en caso de que sea una venta)

5.8.2 Mensajes

El sistema de mensajería manda eventualmente mensajes a los clientes. Estos mensajes están encapsulados en este tipo de objetos, cuyos atributos son:

- Tipo de mensaje. Puede ser de muchos tipos, y depende directamente del tipo de destinatario (puede ser un mensaje ACL, un evento, un cambio en el valor de un título...)
- Contenido. El contenido es una cadena, la cual tiene que ser procesada por el cliente dependiendo del tipo de mensaje que ha enviado.
- Destinatario. El destinatario concreto al que se tiene que enviar el mensaje, un cliente puede tener varios destinatarios (por ejemplo los agentes)

5.8.3 Información

A diferencia de los dos anteriores objetos que son intercambiados por memoria mediante llamadas a métodos remotos, la información se intercambia entre el cliente y el servidor mediante disco. Es decir, al inicio de una simulación, si el cliente detecta que su información no está actualizada se descarga del servidor a disco la información. Esto se hace de forma bastante directa por la implementación que usamos de los WebServices (Axis+Tomcat) pues Tomcat

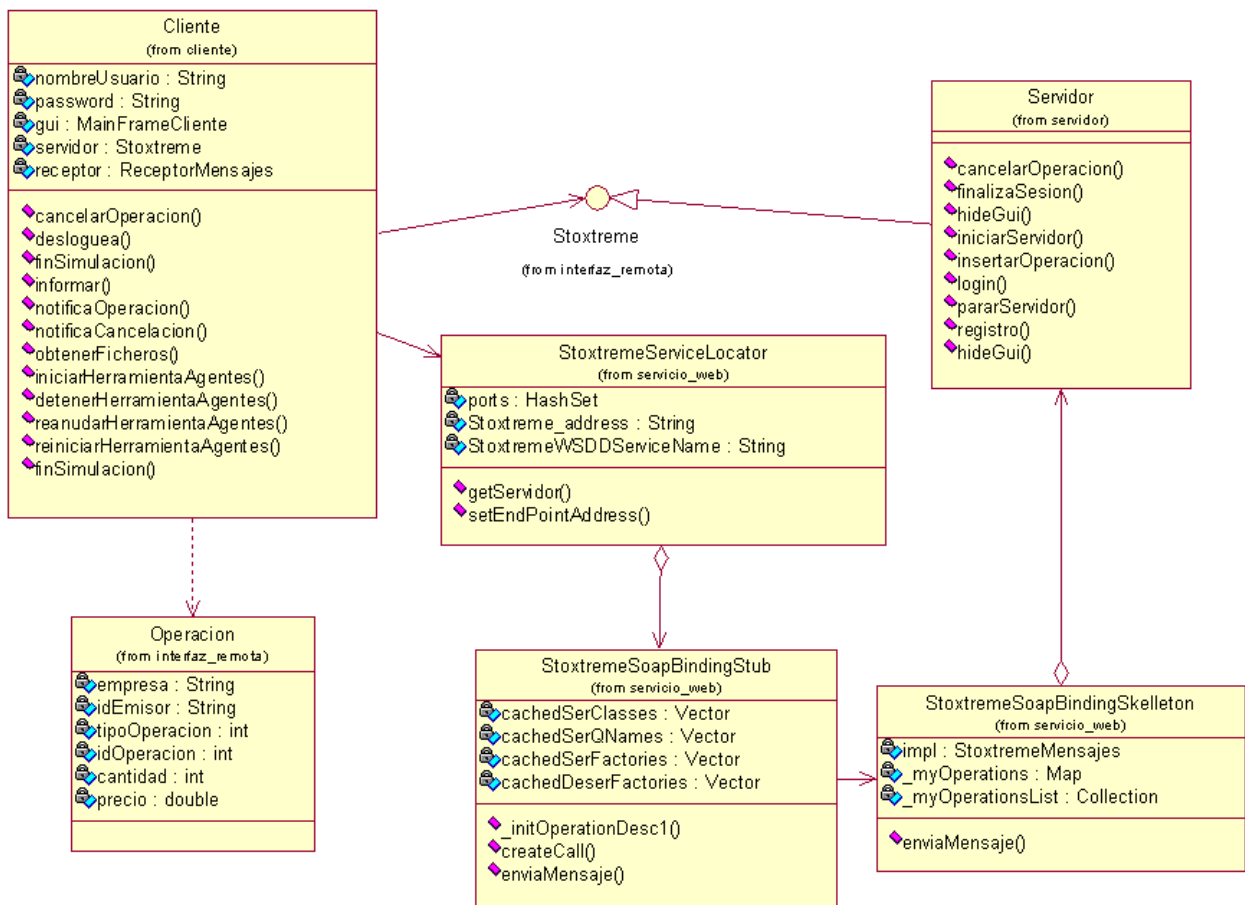
además de ser un servidor de aplicaciones también es un servidor web, lo que nos permite descargar ficheros XML mediante el protocolo HTTP.

Eventualmente el servidor cambia su información, pero estos pequeños cambios en la información se transmiten mediante el sistema de mensajería.

5.9 Implementación de la conexión cliente-servidor mediante Web services

5.9.1 Diagrama de clases de la arquitectura del sistema.

Podemos observar en el siguiente diagrama que el cliente solo conoce del servidor su interfaz remota, que es la implementación mediante Web services. A su vez el cliente necesita un objeto de tipo `StoxtremeServiceLocator` para localizar de forma remota el servicio y poder adquirir el objeto remoto. Una vez adquirido, las llamadas remotas se hacen a través del objeto `Stub` de la parte del cliente y el `Skelleton` de la parte del servidor. El objeto que comparte el sistema es de tipo `Operación`.



5.9.2 Fichero WSDL del servidor

- Tipos Construidos

Definición de los tipos construidos. Definimos el tipo de Operación con sus 5 atributos cantidad, empresa, idEmisor, precio y tipoOperacion.

```
<schema
  targetNamespace="http://interfaz_remota.stoxtreme"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <complexType name="Operacion">
    <sequence>
      <element name="cantidad" type="xsd:int" />
      <element name="empresa" nillable="true"
type="soapenc:string" />
      <element name="idEmisor" nillable="true"
type="soapenc:string" />
      <element name="precio" type="xsd:float" />
      <element name="tipoOp" type="xsd:int" />
      <element name="valido" type="xsd:boolean" />
    </sequence>
  </complexType>
</schema>
```

- Mensajes transmitidos por el Web service.

Mensajes de Request y Response de la operación de registro. El mensaje de request introduce como parámetros el idUsuario y la contraseña.

```
<wsdl:message name="registroRequest">
  <wsdl:part name="idUsuario" type="soapenc:string" />
  <wsdl:part name="password" type="soapenc:string" />
</wsdl:message>

<wsdl:message name="registroResponse">
  <wsdl:part name="registroReturn"
    type="xsd:boolean" />
</wsdl:message>
```

Mensajes de Request y Response de la operación de login. El Request toma como entrada el identificador del usuario y devuelve un entero. Que será -1 si ha ocurrido algún fallo o un entero positivo con el número de empresas del servidor.

```
<wsdl:message name="loginRequest">
  <wsdl:part name="idUserario" type="soapenc:string"/>
  <wsdl:part name="password" type="soapenc:string"/>
</wsdl:message>

<wsdl:message name="loginResponse">
  <wsdl:part name="loginReturn" type="xsd:int"/>
</wsdl:message>
```

Mensajes de Request y Response de la introducción de una operación en la bolsa. El Request toma como parámetros el identificador de usuario al que pertenece esa operación y la operación propiamente dicha. El Response devuelve un entero positivo que es el identificador de la operación dentro de la bolsa.

```
<wsdl:message name="insertarOperacionRequest">
  <wsdl:part name="idUserario" type="soapenc:string"/>
  <wsdl:part name="operacion" type="impl:Operacion"/>
</wsdl:message>

<wsdl:message name="insertarOperacionResponse">
  <wsdl:part name="insertarOperacionReturn"
    type="xsd:int"/>
</wsdl:message>
```

Mensajes de Request y Response para la petición de cancelación de una operación en la bolsa. El Request tiene como entrada el identificador del usuario y el identificador de la operación en la bolsa. No posee Response pues no tiene valor de devolución.

```
<wsdl:message name="cancelarOperacionRequest">
  <wsdl:part name="idUsuario"
type="soapenc:string"/>
  <wsdl:part name="idOperacion" type="xsd:int"/>
</wsdl:message>
```

- Tipos de puerto

Define la operación de registro, y especifica cual es el orden de los parámetros así como cuales son los mensajes que implementan la entrada y la salida.

```
<wsdl:operation
  name="registro" parameterOrder="idUsuario password">
  <wsdl:input
    message="impl:registroRequest"
    name="registroRequest"/>
  <wsdl:output
    message="impl:registroResponse"
    name="registroResponse"/>
</wsdl:operation>
```

Define la operación de login y especifica cuál es el orden de los parámetros así como cuales son los mensajes que implementan la entrada y la salida.

```
<wsdl:operation name="login" parameterOrder="idUsuario password">
  <wsdl:input
    message="impl:loginRequest"
    name="loginRequest"/>
  <wsdl:output
    message="impl:loginResponse"
    name="loginResponse"/>
</wsdl:operation>
```

Define la operación de insertar una operación y especifica cuál es el orden de los parámetros así como cuales son los mensajes que implementan la entrada y la salida.

```
<wsdl:operation name="insertarOperacion"
  parameterOrder="idUserario operacion">
  <wsdl:input
    message="impl:insertarOperacionRequest"
    name="insertarOperacionRequest" />
  <wsdl:output
    message="impl:insertarOperacionResponse"
    name="insertarOperacionResponse" />
</wsdl:operation>
```

Define la operación de cancelar una operación y especifica cuál es el orden de los parámetros así como cuales son los mensajes que implementan la entrada y la salida.

```
<wsdl:operation name="cancelarOperacion"
  parameterOrder="idUserario idOperacion">
  <wsdl:input
    message="impl:cancelarOperacionRequest"
    name="cancelarOperacionRequest" />
  <wsdl:output
    message="impl:cancelarOperacionResponse"
    name="cancelarOperacionResponse" />
</wsdl:operation>
```


- Asociaciones de las operaciones.

Asociación de la operación de login, con su operación correspondiente con el servidor. Asocia la entrada y la salida con las respectivas operaciones de Request y Response.

```
<wsdl:operation name="login">
  <wsdlsoap:operation soapAction="" />
  <wsdl:input name="loginRequest">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

namespace="http://interfaz_remota.stoxtreme"
          use="encoded" />
  </wsdl:input>

  <wsdl:output name="loginResponse">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

namespace="http://interfaz_remota.stoxtreme"
          use="encoded" />
  </wsdl:output>
</wsdl:operation>
```

Asociación de la operación de inserción de una operación, con su operación correspondiente con el servidor. Asocia la entrada y la salida con las respectivas operaciones de Request y Response.

```
<wsdl:operation name="insertarOperacion">
  <wsdlsoap:operation soapAction="" />
  <wsdl:input name="insertarOperacionRequest">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

namespace="http://interfaz_remota.stoxtreme"
      use="encoded" />
  </wsdl:input>

  <wsdl:output name="insertarOperacionResponse">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

namespace="http://interfaz_remota.stoxtreme"
      use="encoded" />
  </wsdl:output>
</wsdl:operation>
```

Asociación de la operación de registro, con su operación correspondiente con el servidor. Asocia la entrada y la salida con las respectivas operaciones de Request y Response.

```
<wsdl:operation name="registro">
  <wsdlsoap:operation soapAction="" />
  <wsdl:input name="registroRequest">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

namespace="http://interfaz_remota.stoxtreme"
      use="encoded" />
  </wsdl:input>

  <wsdl:output name="registroResponse">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
```

Asociación de la operación de cancelar una operación, con su operación correspondiente con el servidor. Asocia la entrada y la salida con las respectivas operaciones de Request y Response.

```
<wsdl:operation name="cancelarOperacion">
  <wsdlsoap:operation soapAction="" />
  <wsdl:input name="cancelarOperacionRequest">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

namespace="http://interfaz_remota.stoxtreme"
      use="encoded" />
  </wsdl:input>

  <wsdl:output name="cancelarOperacionResponse">
    <wsdlsoap:body

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

namespace="http://interfaz_remota.stoxtreme"
      use="encoded" />
  </wsdl:output>
</wsdl:operation>
```

- Definición del servicio

Definimos el servicio con el nombre "StoxtremeService", indicándole cuál es la configuración de las asociaciones y cual va a ser la URL por defecto en la que estará funcionando el servicio web.

```
<wsdl:service name="StoxtremeService">
  <wsdl:port binding="impl:StoXtremeSoapBinding" name="StoXtreme">
    <wsdlsoap:address
      location="http://localhost:8080/axis/services/StoXtreme" />
  </wsdl:port>
</wsdl:service>
```

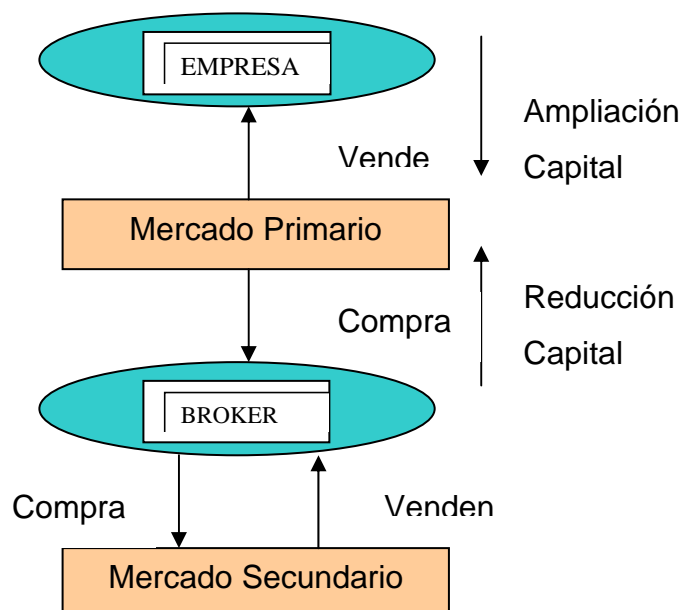
6. DESCRIPCIÓN DEL SERVIDOR

6.1 Tareas del servidor

Para entender mejor la funcionalidad del servidor vamos a ver como se ha definido las funciones de la bolsa real:

1. Intercambio de fondos entre las entidades que necesitan financiación (mayoritariamente empresas) y los inversores. Esta función la realiza el mercado primario o de emisión.
2. Proporcionar liquidez a los inversores. Esto se realiza en el mercado secundario o de negociación.
3. Fijación del precio de los títulos en el mercado.
4. Facilitar información sobre las empresas.
5. Proporcionar confianza a los inversores mediante una garantía jurídica.
6. Publicar los precios de los títulos y las cantidades negociadas.

El servidor básicamente va a tener la misma funcionalidad que proporciona la bolsa en el mundo real, con la única restricción de que sólo vamos a simular el mercado secundario. En vez de simular otro mercado como sería el mercado primario, suponemos una especie de contenedores de las acciones que ningún broker posee, a estos contenedores acude la bolsa para gestionar las compras que no pueden ser atendidas por otras ventas del sistema (en caso de que haya disponibles). El mercado primario no influye directamente en las cotizaciones, en donde de verdad se mueven es en el mercado secundario, donde los agentes de bolsa se intercambian acciones.



Por tanto las funciones que tendrá el servidor serán:

1. Proporcionar a los agentes una serie de interfaces para poder interactuar con el entorno. (Interfaz con los usuarios)
2. Realizar los cálculos matemáticos necesarios para llevar a cabo las fluctuaciones en el precio de las acciones, dependiendo de la oferta y la demanda.
3. Proporcionar la información necesaria a los agentes mediante el sistema de mensajería
4. Informar a los agentes cuando se producen eventos de importancia en la bolsa, para que puedan actuar en consecuencia.

6.2 Parámetros del servidor

Todos estos parámetros tienen un valor por defecto, pero el administrador puede modificarlos desde la interfaz de usuario del servidor.

- Número de empresas: Parámetro que nos indica el número de empresas que entrarán en la simulación.
- Velocidad: Parámetro que nos indica la rapidez o lentitud con que avanza la jornada bursátil, antes de comenzar la simulación el administrador elegirá si la quiere aumentar o disminuir.
- Información Inicial de las Empresas: Indica la ruta del fichero que contiene la información de las empresas que se van a simular.
- Información de usuarios registrados: Indica la ruta del fichero que contiene la información de los usuarios registrados hasta el momento en el sistema.
- Configuración inicial de los agentes: Otro parámetro inicial de la simulación es como queremos que se comporten por defecto los agentes externos que entran en la simulación.

6.3 Interfaz con el administrador

El administrador debe contar con herramientas que le permitan:

- Lanzar eventos que influyen en el comportamiento del mercado, (Ver eventos)
- Variar los parámetros de inicio de sesión.

- Cambiar el comportamiento de los agentes, así como su número.

6.4 Información que emite el servidor

6.4.1 Información en el login

- Valor de las acciones al inicio de sesión. Cuando un usuario se conecta por primera vez al sistema, el servidor envía información sobre los precios de todas las acciones que hay en el mercado. Esta información se guarda localmente en la máquina del usuario, y se va actualizando con sucesivas notificaciones del servidor
- Histórico del precio. El usuario necesita el histórico para saber como ha evolucionado la bolsa hasta la actualidad, esto sirve sobre todo para saber como se han comportado anteriormente los valores y poder invertir en consecuencia.
- Información de las Empresas. Cada empresa tendrá asociada información para inversores para que los usuarios dispongan de información útil, aunque esta información no es definitiva con respecto a su comportamiento.

6.5 Sistema de eventos

Como ya se ha comentado, el precio de las acciones viene determinado por las operaciones que realizan los agentes, el problema es que es muy difícil determinar cómo se van a comportar en la realidad los agentes de bolsa a la hora de comprar y vender acciones. Para facilitar la simulación y con objetivo de que el usuario pueda prever situaciones se va a introducir la funcionalidad de eventos en el servidor. Un evento es un hecho imprevisible que puede afectar al clima bursátil y a las valoraciones que hacen los agentes de bolsa respecto a una determinada acción. Estos eventos son comunicados a los agentes y repercutirá directamente en el precio de las acciones.

Va a haber dos tipos de eventos:

- Automáticos. Estos eventos se producen automáticamente en la bolsa sin la intervención del administrador.

- **Manuales.** Son eventos introducidos por el administrador para ver como se comporta la simulación cuando suceden y poder predecir cual es la mejor estrategia a seguir.

Los eventos se definirán en forma de condición acción para que para el administrador sea más fácil su introducción. Los automáticos estarán predefinidos en el sistema, y su representación interna será de la misma forma que los manuales, aunque el administrador no podrá cambiarlos.

Un ejemplo del mundo real sería si el administrador quiere que cuando el precio de Endesa llegue a un valor M, Endesa ante sus buenas perspectivas de futuro lance una OPA sobre Gas Natural. El evento resultaría algo así:

SI (Endesa.Valor > M) **ENTONCES** Evento(OPA(Endesa, GasNatural))

Este evento se comunicara a todos los usuarios del sistema y estos actuarán en consecuencia, lo normal sería que el precio de las acciones subieran pues cuando se realiza una OPA significa que Endesa va a hacer muchas compras y a los inversores les gusta eso.

Los tipos de eventos automáticos serán:

- La cotización de una acción alcanza el precio máximo o mínimo y se suspende su cotización.
- Eventos ante el cambio en el periodo de cotización (apertura y cierre de la bolsa)

Los tipos de eventos manuales serán:

- Ampliaciones y reducciones de capital
- Aumento ó disminución de los tipos de interés.
- Aumento ó disminución de la inflación
- Eventos específicos respectivos a empresas (se referiría a por ejemplo si una empresa tiene que afrontar una inspección de hacienda, rumores de quiebra, etc...)
- Decisiones políticas que hagan crecer el producto interior bruto (PIB) ó que favorezcan a los inversores tendrán efecto positivo en las bolsas.

- Otros eventos. En este tipo de eventos habría que especificar directamente el efecto que podría tener sobre la bolsa. Por ejemplo si el administrador quiere introducir un evento al respecto de que hay una guerra en Irak, tendría que informar a los agentes de que va a influir negativamente en el precio de todos los sectores en los que es crítico el precio de la gasolina.

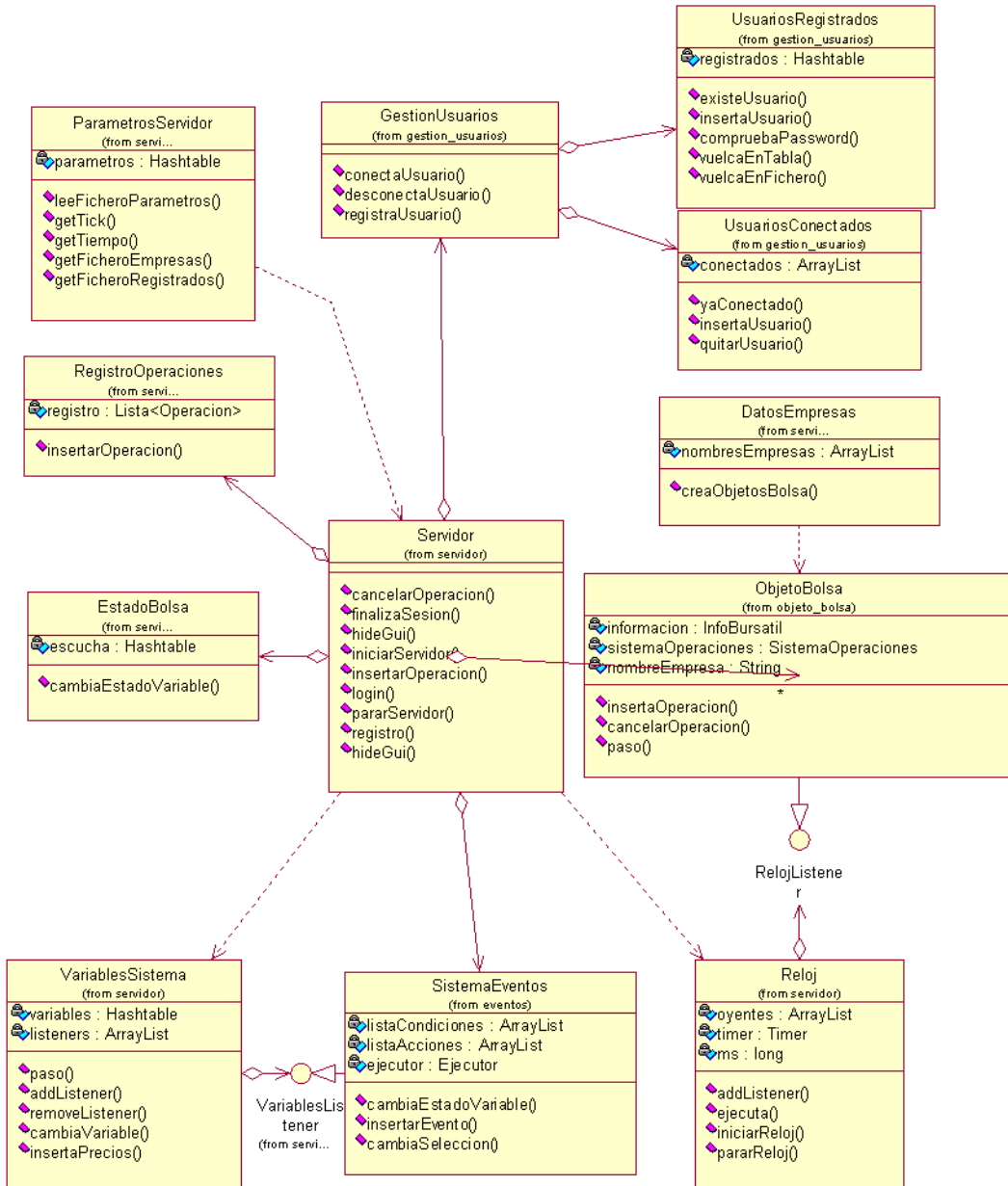
6.6 Arquitectura del servidor

A continuación vamos a relacionar la funcionalidad del servidor explicada anteriormente con la arquitectura del mismo.

Como hemos visto, el servidor tiene varias funciones:

- Proporcionar al administrador una serie de **interfaces** para variar las simulaciones.
- Realizar los cálculos matemáticos necesarios para llevar a cabo las **fluctuaciones** en el precio de las acciones.
- Proporcionar la **información** necesaria a los agentes mediante el sistema de mensajería.
- Informar a los agentes cuando se producen **eventos** en la bolsa.
- Gestionar la conexión de los clientes a la bolsa.

Estas funcionalidades están bien separadas en el diseño del servidor, como se puede ver a continuación:

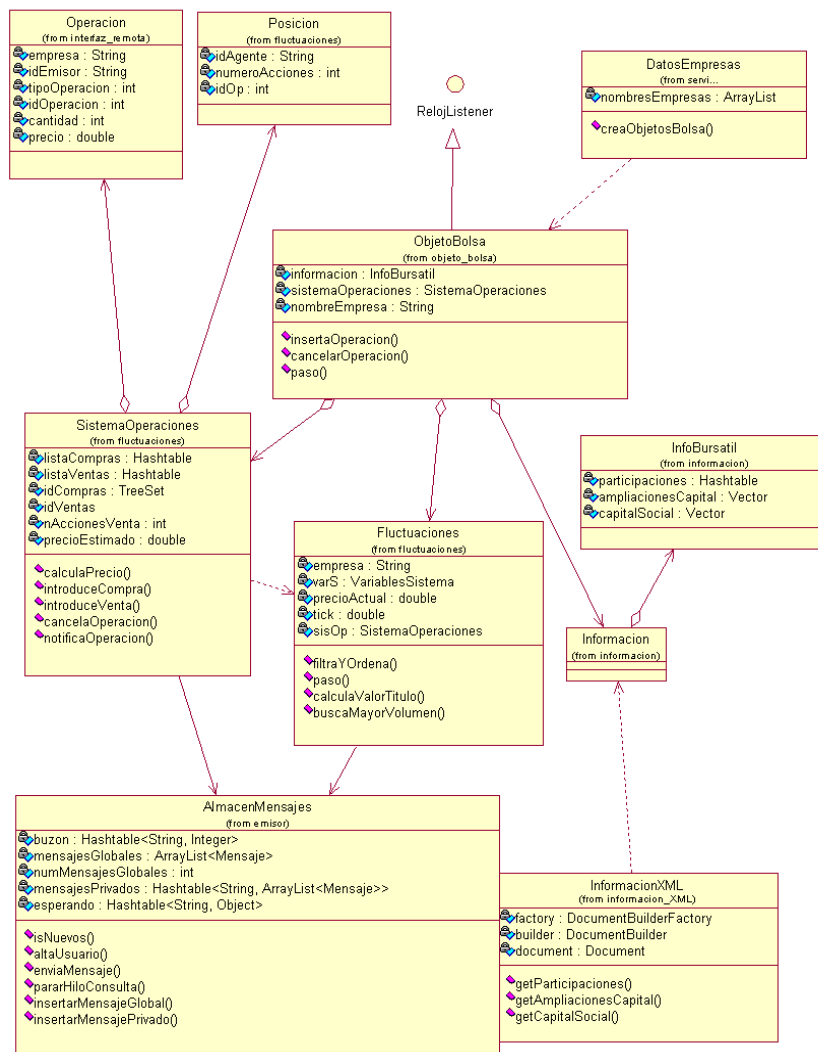


6.6.1 ObjetoBolsa

Es el que se ocupa del tratamiento de las acciones. Hay un objeto bolsa para cada empresa que cotiza en la bolsa.

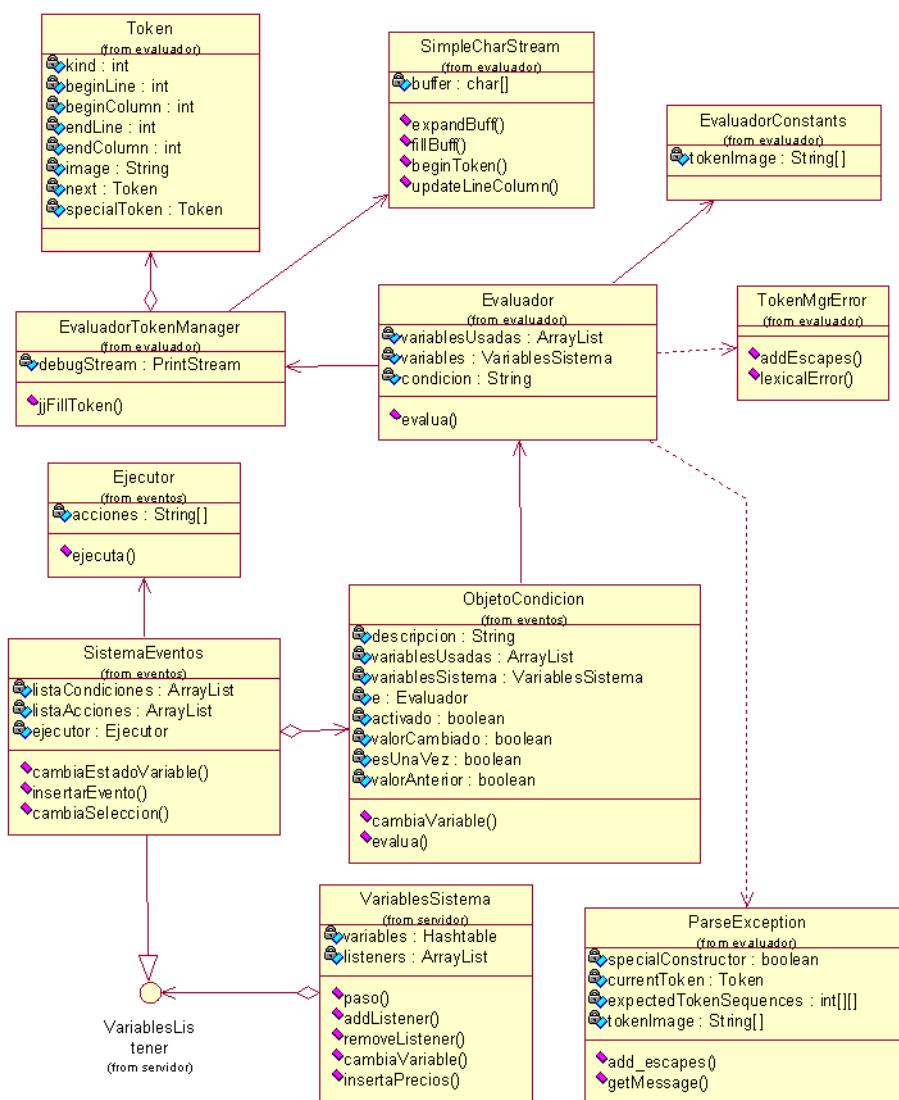
La información de las empresas está almacenada en ficheros XML, por lo que el objeto bolsa incorpora un parser para leer dicha información al iniciar la simulación y almacenarla en si mismo, para que el acceso resulte mas eficiente.

El objeto bolsa también tiene un sistema de operaciones en el cual se almacenan las peticiones de los usuarios, que se cruzan periódicamente, calculándose así es nuevo precio de las acciones. Este sistema también se encarga de enviar al sistema de mensajes los resultados del cruce, para que notifique a los usuarios correspondientes.



6.6.2 Sistema de eventos

Se ocupa del tratamiento de los eventos introducidos por el administrador. Cuenta con un evaluador implementado en JavaCC que comprueba periódicamente si las condiciones del lanzamiento del evento se cumplen. Cuando esto ocurre, el ejecutor se ocupa de llevar a cabo la acción asociada al evento (normalmente emitir un mensaje global con determinada información).



6.6.3 Gestor de usuarios

Este subsistema permite a nuevos usuarios registrarse, conectarse y desconectarse al/del sistema para participar en la simulación. También contiene un pequeño parser, pues la relación de nombre de usuario y contraseña de los usuarios ya registrados se guarda en un fichero XML, para que perdure cuando se cierre el servidor.

6.6.4 Interfaz gráfica para el administrador

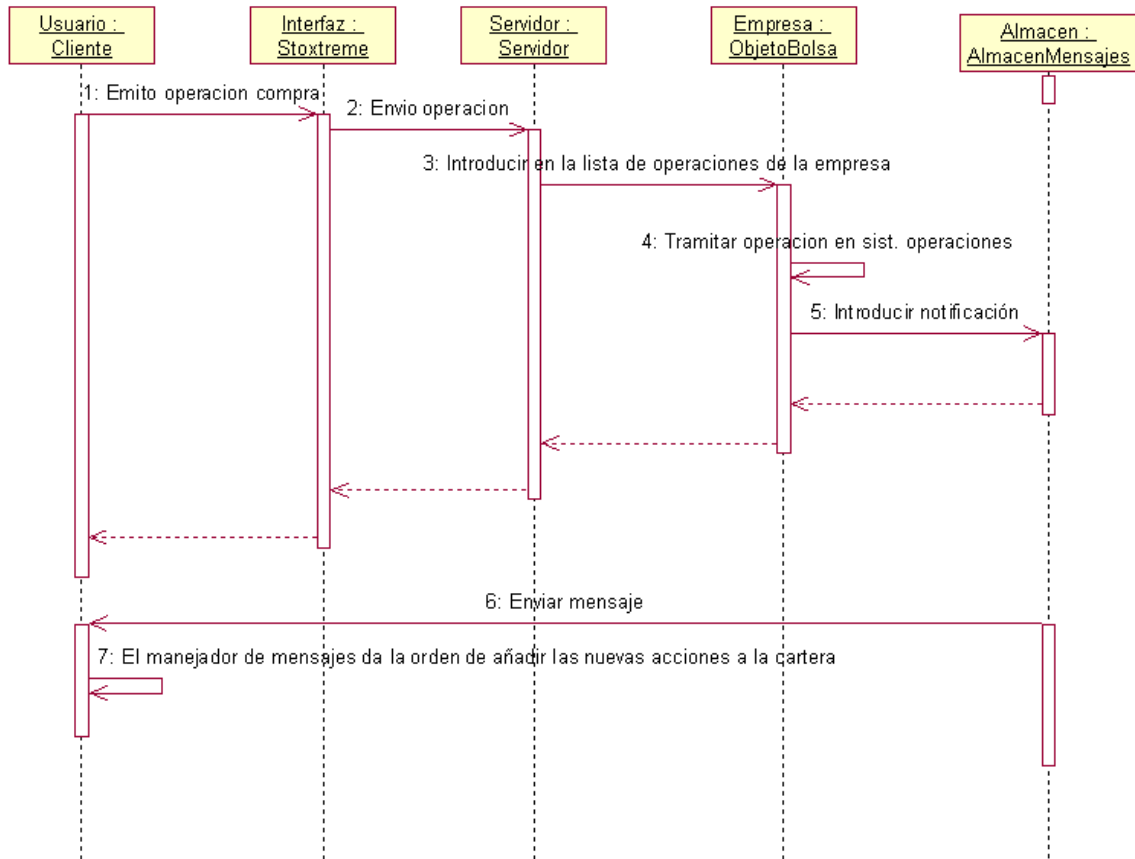
Para poder llevar un control de la simulación y variar los parámetros en base a los que esta se realiza, es necesario que el administrador disponga de una interfaz gráfica que le facilite la tarea.

Mediante la interfaz que hemos implementado, el administrador puede ver las operaciones que entran en el sistema, la evolución de los precios de las acciones, los usuarios registrados y conectados, modificar los parámetros de comportamiento de los agentes, e introducir eventos en la bolsa.

6.6.5 Interfaz para los usuarios

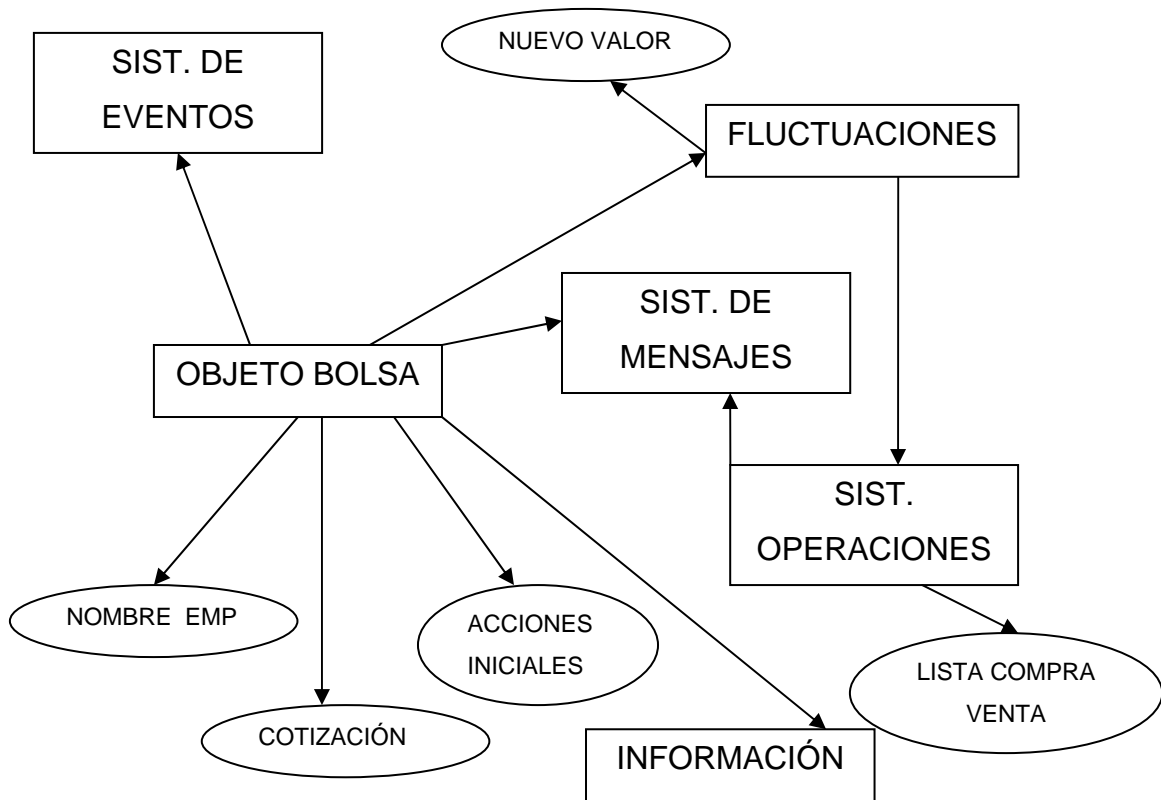
El servidor también proporciona una interfaz para que los usuario accedan de manera remota a los servicios del servidor, lo que les permite registrarse y conectarse a la simulación, y participar en la bolsa introduciendo acciones de compra/venta.

En el siguiente diagrama de secuencia se muestran los pasos que se dan desde que se introduce una operación de compra en el sistema hasta que ésta se tramita y se informa al usuario que la introdujo.



7. FLUCTUACIONES DEL SISTEMA

La representación de los valores de las empresas en el sistema bursátil se ha realizado de la siguiente forma:



Cada objeto bolsa corresponde a una empresa distinta del IBEX 35, cada objeto está en contacto con el sistema de eventos y el de mensajes para poner en conocimiento de todos los agentes y usuarios la situación actual de la empresa en el mercado continuo. Cada empresa tiene asociada a ella su información, descripción de sus movimientos y situación.

Todos los objetos bolsa contienen además un sistema de operaciones. El sistema de operaciones tiene 2 tablas hash (una de compras y una de ventas) que registran las operaciones pendientes sobre las acciones de ese objeto bolsa. Las tablas están ordenadas por precios, cuando un agente o usuario envía una operación se pone a la espera en una lista de prioridad asociada a cada precio.

Cada cierto tiempo (Parámetro introducido por el administrador al inicio de la simulación) el sistema calcula el nuevo precio mediante el siguiente algoritmo:

7.1 Algoritmo de selección del nuevo precio

- Partimos del precio antiguo del valor.
- Existe un rango de valores por encima y por debajo de dicho precio en las cuales se han posicionado distintos compradores y vendedores (este rango dependerá del tick).
- Elegiremos como nuevo valor el precio al que mayor número de títulos se negocien, es decir, dónde más coincidan vendedores y compradores.
- En el caso en que dos valores coincidan escogeremos el que menor desequilibrio produzca en el mercado.
- En el caso de que haya dos valores que también coincidan deberemos elegir el más próximo al precio del cierre del día anterior.
- En el caso que no haya acciones de venta, el sistema de operaciones mira a ver si quedan acciones en el objeto bolsa para comprar.

Una vez obtenido el precio nuevo mediante el anterior algoritmo, se realizan los cruces eliminando las peticiones atendidas de sus correspondientes tablas hash e informando a los usuarios implicados de que sus peticiones han sido atendidas mediante el sistema de mensajes. El sistema de mensajes también informa periódicamente a todos los usuarios de los cambios que se han producido en los precios de las empresas.

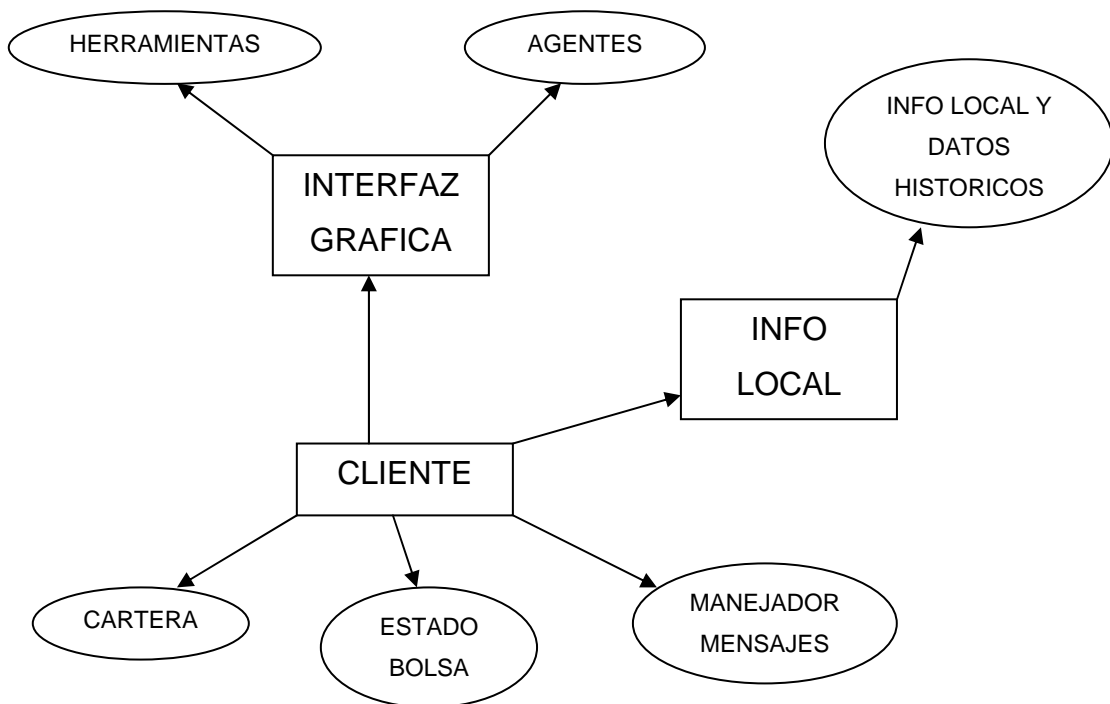
8. DESCRIPCIÓN DEL CLIENTE

Este módulo consta de una interfaz gráfica de interacción del usuario con el sistema el cliente mediante este medio podrá realizar las operaciones que desee sobre las acciones cumpliendo ciertas restricciones.

Para permitir dicha comunicación es necesario que el usuario pueda estar en contacto con el sistema de mensajes, a partir de él también podrá estar informado de los cambios que se produzcan en tiempo real a parte de pedir información que necesite en cualquier momento.

También podrá observar al resto de agentes que interactúan con él en la simulación bursátil.

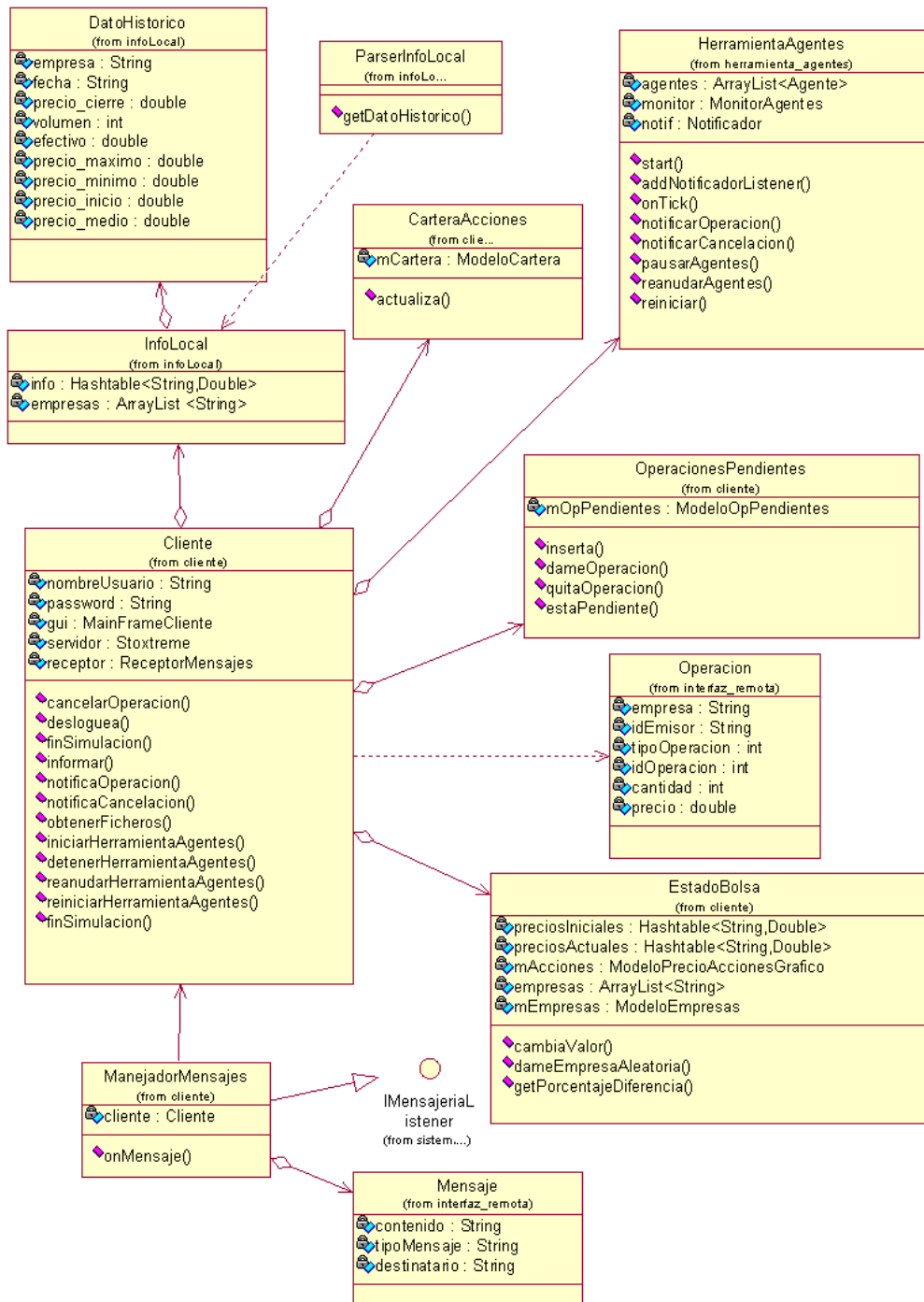
Este sería el esquema general del cliente ahora profundizaremos cada punto de su diseño.



8.1 Arquitectura del cliente

Los clientes son imprescindibles para la simulación, pues son ellos los que hospedan los sistemas de agentes que van a intervenir en la bolsa para que su comportamiento sea relativamente real.

Esta es la estructura en clases de nuestro cliente:



Para poder acceder a la simulación, los clientes tienen que identificarse en el servidor. Para ello necesitan haberse registrado previamente, eligiendo un nombre de usuario y una contraseña.

Cuando se conectan al servidor, éste les envía información sobre las empresas en ficheros XML. Esta información se extrae mediante un parser y se guarda en un objeto **InfoLocal**. La excepción son los datos históricos, estos se mantienen en el XML y se extraen individualmente cuando se soliciten, guardándose temporalmente en un objeto de tipo **DatoHistorico**. Esto es así porque los datos históricos son demasiado grandes, guardarlos todos en un objeto implicaría un gasto de memoria considerable.

El cliente puede intervenir en la simulación y comportarse como un agente más, por lo que dispone de las opciones de emitir y cancelar operaciones de compra/venta. También dispone de una cartera de acciones (**CarteraAcciones**) en la que se almacena las acciones que ha ido adquiriendo, y el precio al que se adquirieron. En el objeto **OperacionesPendientes** se almacenan las operaciones que aun no han sido atendidas por el servidor.

El usuario recibe información de la bolsa mediante el sistema de mensajes. En el **ManejadorMensajes** se reciben los mensajes del servidor a través de la interfaz **IMensajeriaListener** y se ejecuta la acción correspondiente en el cliente (actualizar precios del **EstadoBolsa**, eliminar una operación pendiente, notificar a un agente una operación pendiente...).

El objeto **EstadoBolsa** contiene una copia del estado de la bolsa en cada momento. Es útil para que los agentes tomen sus decisiones y para que el usuario pueda visualizar las fluctuaciones de los precios.

Por supuesto, el usuario cuenta con una interfaz gráfica que le facilita la interacción con el sistema. Ésta se verá en profundidad en la sección del manual de usuario.

8.2 Interacción del cliente con la herramienta de agentes

El usuario podrá generar el número de agentes que desea introducir al sistema para dar mayor realismo a la simulación, cuantos más agentes más real se comportará el sistema, además el usuario podrá observar las notificaciones y los mensajes que produzcan los agentes. Las estrategias que siguen los agentes serán asignadas automáticamente en función de los parámetros que haya definido el administrador. Una vez arrancado el sistema de mensajes, el usuario podrá eliminar manualmente determinados agentes, reiniciarlos o parar el sistema de agentes.

9. SISTEMA MULTIAGENTE

Una de las principales características de nuestro sistema es el sistema multiagente que posee, gracias al cual se pueden modelar los comportamientos de la bolsa. Al finalizar la aplicación hemos conseguido lo que creemos es una estructura del sistema multiagente muy completa pero quizás nos ha faltado agentes más complejos y por eso el comportamiento general del sistema no es todo lo bueno que se desearía.

9.1 Características del sistema multiagente

- Comunicación local entre agentes basada en el estándar de FIPA ACL.
Los agentes poseen comunicación local y se intercambian mensajes siguiendo el protocolo FIPA. Los comportamientos establecen que tipo de mensajes y ante qué eventos se mandarían esos mensajes.
- Percepción por parte de los agentes de los eventos en bolsa.
El cliente a través del manejador de mensajes informa al notificador de los agentes. Siguiendo una arquitectura de modelo-controlador, se actualiza uno de los componentes de los agentes que es el perceptor.
- Ejecución multihilo de los agentes.
Cada agente es un hilo que toma sus decisiones de forma concurrente al resto de los agentes. Una vez tomada la decisión se manda al monitor de los agentes que es donde se van a serializar las operaciones hacia el entorno.
- Población dinámica de agentes dependiente de sus beneficios.

Eventualmente los agentes saldrán o entrarán en la simulación dependiendo de las características del entorno.

- Configuración remota de la población (Configuración global del sistema multiagente).

El administrador del sistema podrá definir las características globales del sistema y los modelos de comportamientos generales de los agentes.

- Control local de la herramienta de agente, mediante interfaz gráfica (Configuración de agentes específicos y sobre comportamientos concretos).

El usuario podrá definir los detalles del contenedor que posee su cliente, tales como el número de agentes, la velocidad de ejecución, etc...

- Interacción por parte del usuario con los agentes. Además de la configuración del contenedor local de agentes, el usuario puede borrar y comprobar las operaciones de agentes concretos.

9.2 Tipos de agentes

Antes de hacer una taxonomía de los tipos de agentes en el sistema, hay que tener en cuenta que ninguno de estos tipos es absoluto, más bien se mezclan varios comportamientos y resultan en agentes que se comportan más o menos de la siguiente forma.

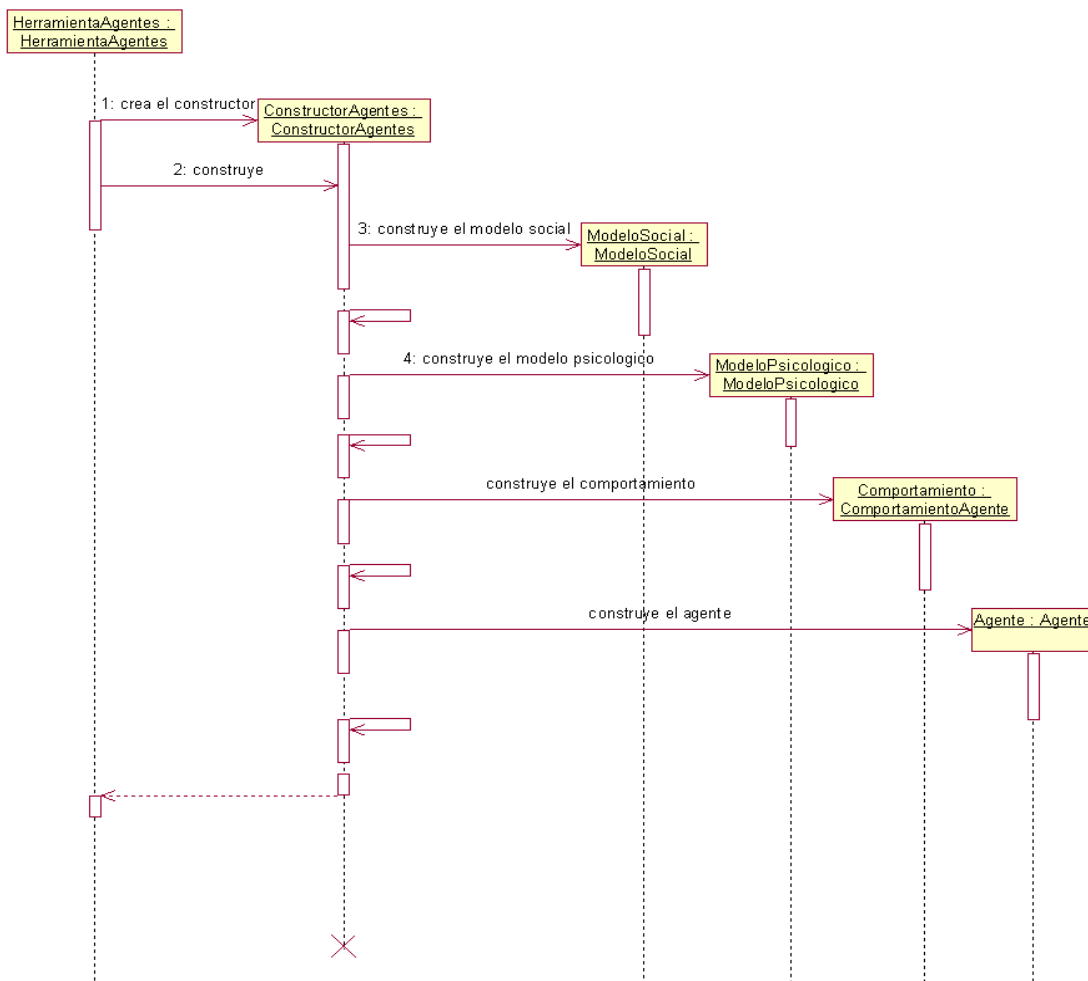
Por ejemplo, aunque un agente sea informador, puede tener un factor social en la forma en la que reproduce esa información y comportarse de manera aleatoria al comprar y vender acciones.

- Agentes aleatorios. Introducidos para producir el ruido aleatorio típico en este tipo de un sistema tan estocástico como es la bolsa.
- Agentes informadores. Proporcionan consejos al resto de los agentes. Hay varias subcategorías.
 - Analistas financieros. Analizan los datos bursátiles y hacen recomendaciones a los agentes que se las piden, cobrando por esas recomendaciones.
 - Agentes del tipo “publicitarios”. Los publicitarios informan a todos los agentes por igual, los agentes se dejan influir en mayor o menor medida por la publicidad.
 - Agentes introductores de rumores y de eventos. Estos dos subtipos

se comportan de forma similar. Básicamente introducen una indicación a los agentes de que va a bajar o subir la bolsa. (En caso de los eventos serán los agentes los que interpretan si va a bajar o subir). Los agentes harán caso a este rumor en función de lo que se “fíen” de los agentes.

- Agentes inteligentes. Contienen una serie de reglas las cuales están especificadas como un sistema clasificador. Evolucionan según les vaya mejor o peor en la bolsa aprendiendo de sus errores.
- Agentes sociales. Poseen conocimiento en forma de redes sociales, intercambian información con los conocidos y pueden retransmitirla.

El siguiente diagrama de secuencia ilustra los pasos que se dan para crear a un agente.



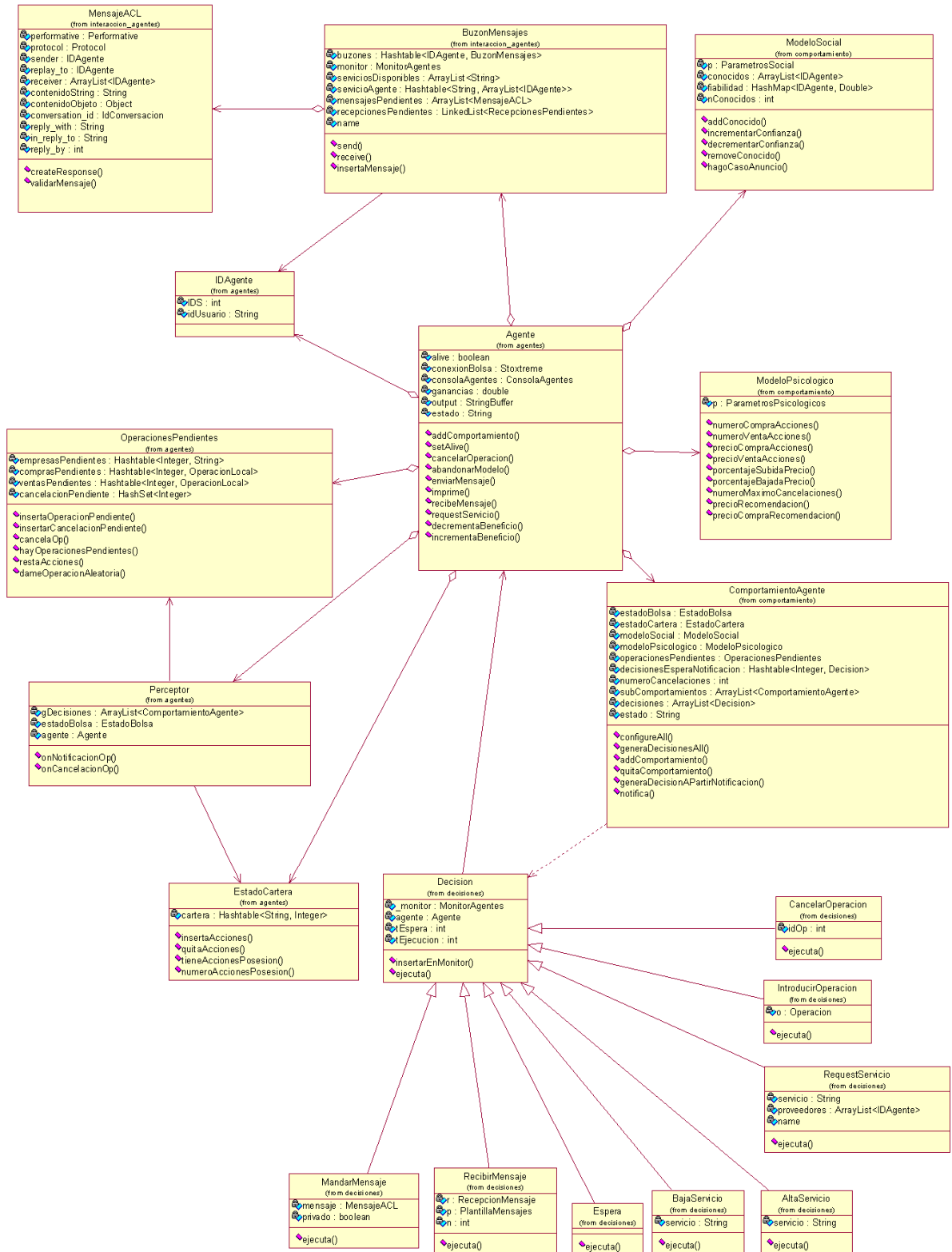
9.3 Estructura general de los agentes

Los agentes están divididos en tres partes totalmente diferenciadas que corresponden con las indicadas en la mayoría de los libros sobre teoría de agentes:

- Toma de decisiones. Analiza los datos que posee el agente y genera decisiones las decisiones oportunas de acuerdo con la estrategia que implementa. Estas decisiones se pueden tomar por el comportamiento específico o por los subcomportamientos anidados que posea.
- Ejecución de acciones. Sistema centralizado para la ejecución serializada de todas las acciones llevadas a cabo por los agentes decide cuando un agente puede ejecutar la decisión, pero las acciones propiamente dichas las realiza directamente el agente.
- Percepción del mundo. Los perceptores se encargan de actualizar las estructuras sobre las que los agentes toman las decisiones ante eventos que ocurren en la bolsa y a partir de las decisiones de otros agentes. Esta percepción esta implementada siguiendo un patrón observador, de forma que cuando hay un cambio en el mundo se actualiza el perceptor.

Observamos en el siguiente diagrama de clases como está compuesto un agente. Podemos observar en el centro el Agente, que está compuesto por varios objetos mediante relaciones de agregación. Entre ellos los modelos, el comportamiento, el perceptor y los estados de la bolsa y de la cartera. Además posee un buzón de mensajes que es el que almacena sus mensajes ACL correspondiente (explicado en una sección posterior). El comportamiento genera las decisiones correspondientes las cuales extienden todas de una clase común que es decisión. Una vez que el monitor decida ejecutar la correspondiente acción esta ejecuta la correspondiente acción del agente. El perceptor básicamente modifica los estados de la bolsa, de la cartera y las operaciones pendientes.

Herramienta de simulación bursátil sobre agentes software y web services

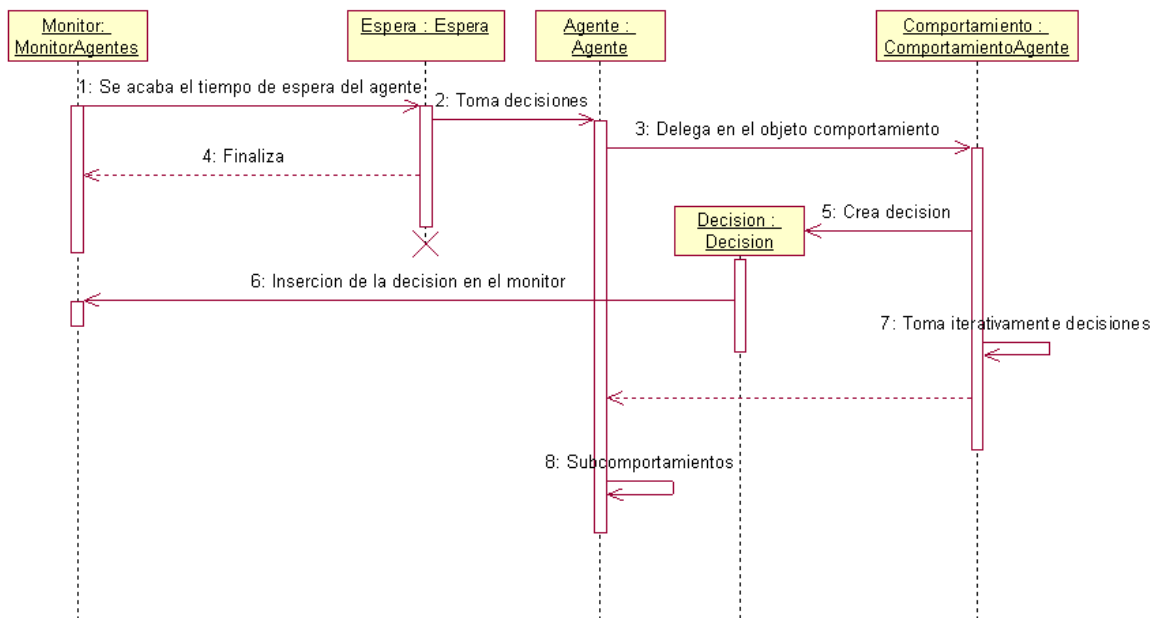


9.4 Toma de decisiones

El “cerebro” de los agentes una estructura interna que llamamos “Comportamiento”.

El objeto Comportamiento, es una clase abstracta la cual sirve para implementar múltiples comportamientos. Se encarga de generar todas las decisiones y llevar un control sobre las operaciones efectuadas. Un Comportamiento puede tener anidados varios subcomportamientos para ampliar la funcionalidad de un único objeto de ese tipo. Estos subcomportamientos también se pueden insertar y retirar en tiempo de ejecución de forma que puede variar el comportamiento de un agente dependiendo del estado de la bolsa.

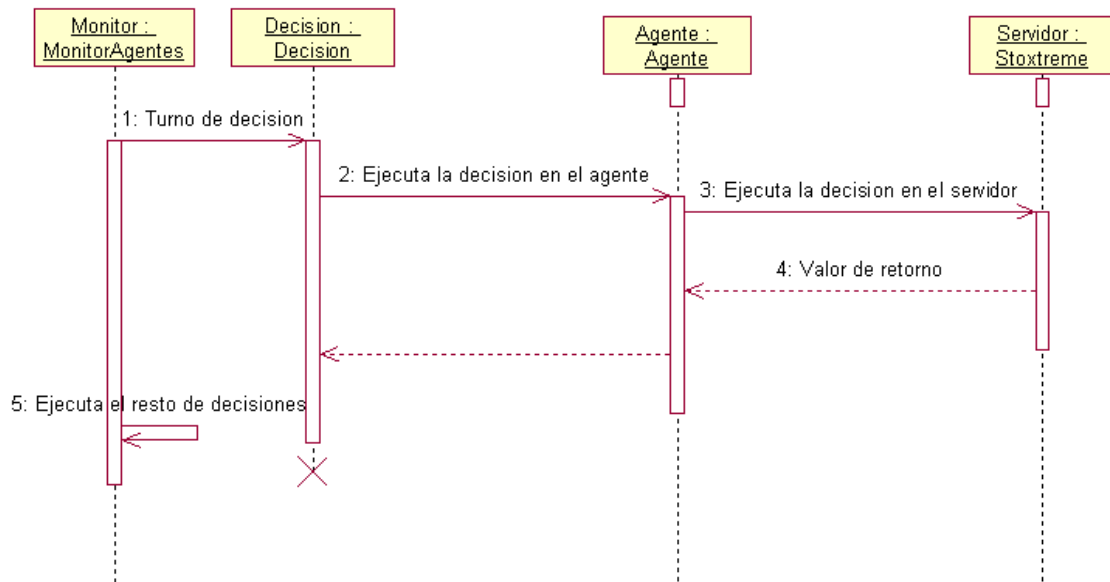
Estas decisiones pueden basarse (o no) en los modelos social y psicológico del agente. Estos modelos social y psicológico están desarrollados a partir de los ficheros de configuración remotos que posee el servidor.



9.5 Ejecución de acciones

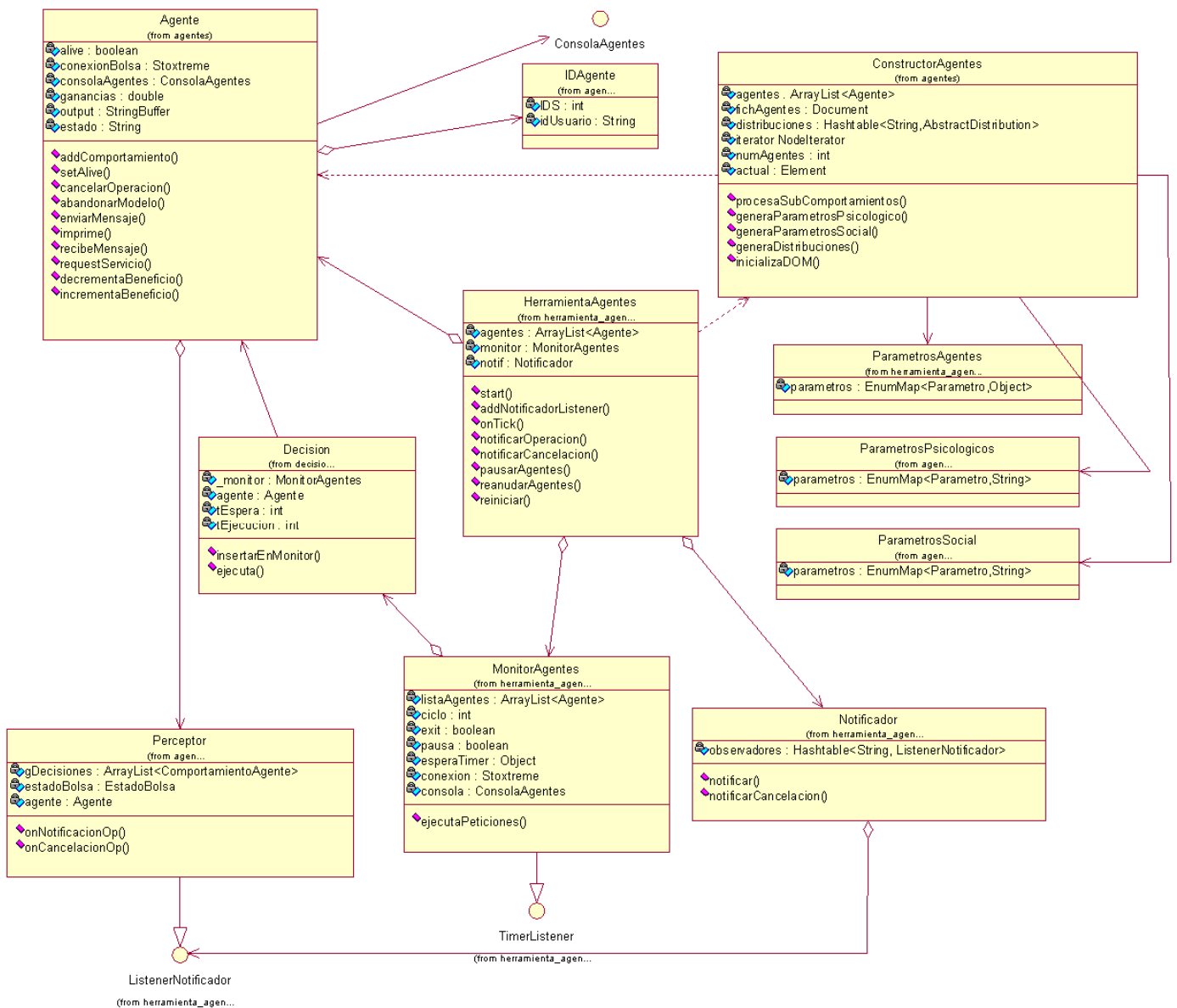
Hemos implementado un monitor, con un modelo productores-consumidor. Los agentes esperan inactivos hasta que el monitor los despierta para que tomen decisiones o ejecuten las acciones que tenían planeadas. Una vez tomadas las decisiones se las pasan al monitor que es el que se encargará de avisarles cuando las tengan que ejecutar.

El monitor también se encarga de proporcionar una interfaz a los agentes para la interacción con la bolsa.



9.6 Estructura del sistema de agentes.

En el siguiente diagrama podemos observar como es la estructura de clases del monitor del sistema de agentes. En el centro está la HerramientaAgentes que es la clase principal del sistema, crea los agentes por medio del ConstructorAgentes que usa los parámetros necesarios para la construcción. Los agentes mandan las decisiones al MonitorAgentes y les notifican las operaciones por medio del Notificador y a través del Perceptor.



9.7 Interacción agente-bolsa

La bolsa en nuestro sistema se encuentra de forma remota implementada mediante *Web services*. Desde el punto de vista de los agentes, la implementación con *Web services* es totalmente transparente. Esto es debido a que el monitor le proporciona una interfaz que es a la que el propio agente tiene acceso. Gracias a esta interfaz podemos asegurar que aunque haya muchos agentes, no todos van a realizar sus peticiones de golpe, sino que se van a ejecutar una a una.

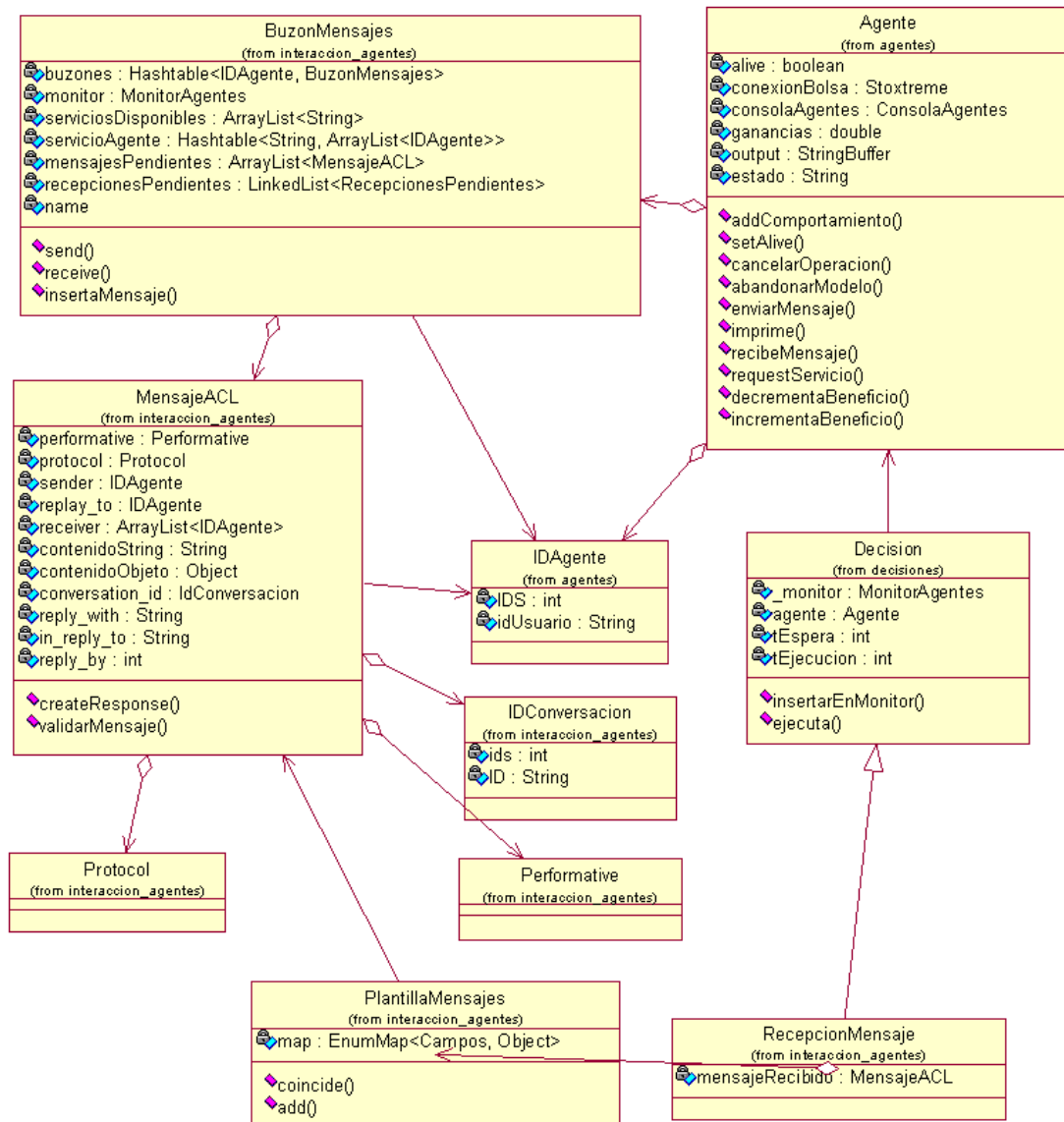
9.8 Comunicación entre los agentes

Hemos realizado una implementación del estándar ACL definido por la FIPA (www.fipa.org) Seguimos las especificaciones tanto de la estructura de los mensajes ACL como la definición de los protocolos y los actos de comunicación definidos en este estándar.

La utilización de los estándares FIPA facilita mucho la utilización de nuestro sistema por desarrolladores que deseen implementar agentes con funcionalidad adicional, al ser un protocolo bien conocido.

Este protocolo basa su comunicación en actos del lenguaje, de forma que un agente manda un mensaje a otro con uno de estos actos de comunicación. Además si espera mantener una conversación con el agente, puede enmarcar esta comunicación dentro de un protocolo de comunicación de forma que tanto el receptor como el emisor saben cuales y cuantos son los mensajes que se van a intercambiar entre ellos.

En el siguiente diagrama de clases podemos observar como hemos realizado la implementación del sistema de comunicación entre agentes. Cada agente posee un identificador uno en todo el sistema y un buzón de mensajes que es donde se van a almacenar los mensajes recibidos. Cuando un agente desea recibir un mensaje todo lo que debe hacer es crear un objeto del tipo abstracto *RecepcionMensaje* que es una abstracción de la ejecución de una operación ante la recepción del mensaje. Además tiene que rellenar el objeto de *PlantillaMensaje* de forma que cuando un mensaje encaje con esta *Plantilla* se ejecute la acción definida por la recepción del mensaje.

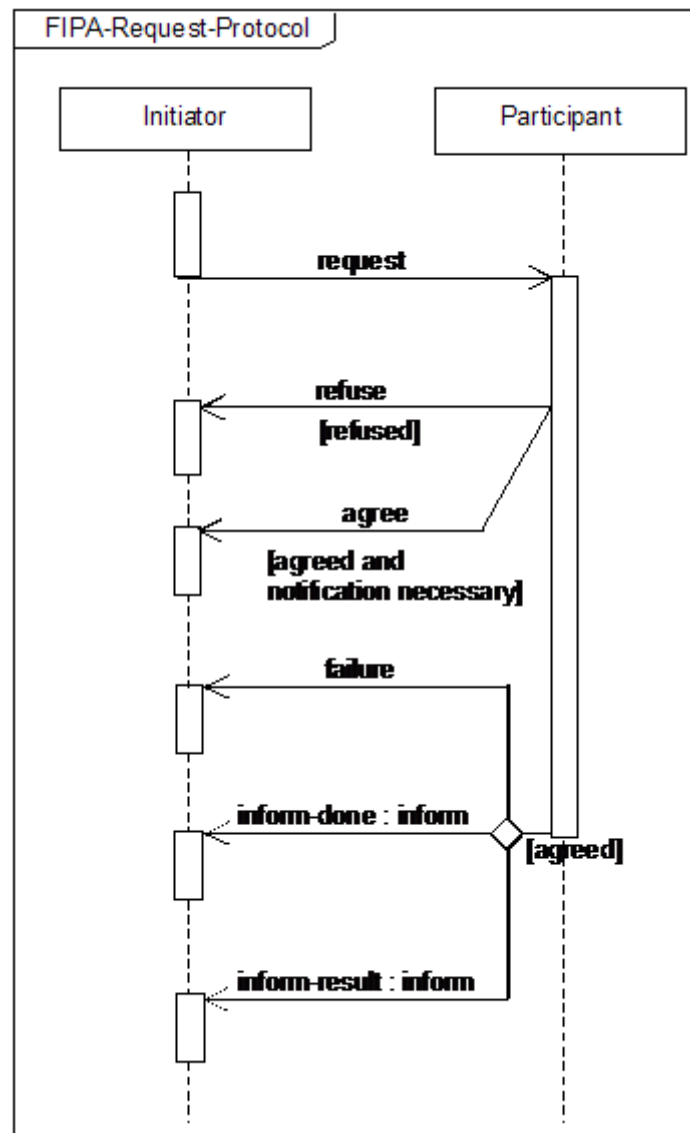


9.9 Agentes informadores y mensajes intercambiados del protocolo ACL

En nuestro sistema hemos implementado básicamente dos tipos de agentes que hacen uso de los mecanismos de comunicación mediante mensajes ACL. Dos emisores y otros dos receptores lo que hacen un total de cuatro tipos de comportamientos distintos.

El primer tipo es la comunicación de rumores. Puede haber dos fuentes de un rumor, o bien un evento del sistema o bien otro agente que transmite un rumor. Los rumores están identificados por una palabra clave "RUMOR", una empresa sobre la que trata un rumor y una potencia. La potencia del rumor es un porcentaje que es lo que se espera que suba o que baje, de forma que si es una potencia de 2.0 es que se espera que suba un 200%. Cuando el sistema de eventos es el que difunde el rumor, el rumor llega por igual a todos los agentes mediante un mensaje de BROADCAST. Cuando llega un rumor, los agentes retransmiten el mismo hacia sus conocidos y lo atenúan otro porcentaje, de forma que cuando está lo suficientemente próximo a 1 el rumor desaparece del sistema.

El segundo tipo de comportamiento que tiene comunicación es una petición de recomendación. Hay dos tipos de agentes que efectúan recomendaciones sobre empresas de la bolsa: análisis fundamental y análisis técnico. Cuando un agente no está seguro o quiere asegurarse de realizar una buena inversión tiene que pedir una recomendación a uno de estos agentes. El proceso es el siguiente. Los agentes que realizan las recomendaciones tienen que dar de alta sus respectivos servicios en el sistema de mensajes, después los agentes que desean pedir una recomendación solicitan al sistema de mensajes los agentes que proveen del servicio. Una vez el agente solicitante tiene los identificadores de los agentes que pueden proveer de ese servicio elige uno y empieza entonces una comunicación que sigue el protocolo ACL-REQUEST, como la que se especifica en el siguiente diagrama. Siendo el iniciador el solicitante de la petición y el participante el agente que provee de la recomendación. Una vez que se ha intercambiado el mensaje inform, el sistema retirará dinero al agente solicitante de forma que se pague al agente recomendador.



9.10 Población dinámica de agentes

Eventualmente los agentes podrán abandonar la bolsa si observan que la cantidad de dinero que están perdiendo supera lo que esperaban, y también dependiendo de lo bien o mal que vaya la bolsa.

Esto lo hemos introducido debido a un comportamiento que es observable en la bolsa real, y es que en épocas que la bolsa está en alza la gente suele invertir mucho más en bolsa, pues es un valor que proporciona más dinero que las inversiones a plazo fijo. En cambio, en épocas en las que la bolsa está en baja, los inversores suelen desplazarse de la bolsa a otro tipo de inversiones más seguras.

9.11 Configuración de agentes

Como hemos comentado anteriormente, hay dos tipos de configuraciones. Una remota y otra local, a continuación vamos a aclarar las principales diferencias.

- Configuración remota. Se trata de una configuración global del sistema, pues define los tipos de agentes (en relación a su comportamiento) y las proporciones de agentes que habrá en el sistema global. Esta configuración estará definida mediante ficheros estáticos de configuración que son cargados al inicio de la simulación y descargados por los usuarios que conectan con el servidor.
- Configuración local. Al contrario que la configuración remota, la configuración local es específica de cada agente, pues de lo que se encarga es de ajustar los comportamientos de agentes concretos en el sistema. Es una configuración dinámica que el usuario realiza mediante interfaz gráfica.

9.12 Ficheros de configuración

El fichero de configuración de los agentes, es definido remotamente por el administrador del sistema.

Este fichero de configuración consiste en un fichero de XML con la siguiente estructura:

- Parámetros generales. Son los valores del sistema en general.
 - Número máximo y mínimo de agentes. Límites dentro del parámetro que tiene que introducir el usuario del número de agentes en el sistema.
 - Tiempo de espera mínimo y máximo. Límites para el tiempo de espera dado que el usuario es el que debe introducir el parámetro.
 - Gasto máximo de cada agente. Gasto ante el cual el agente se saldrá de la bolsa.
 - Ratio de respawn. Probabilidad de que un agente que haya salido de la bolsa vuelva a entrar en ella.
 - Atenuación del rumor. Porcentaje que un rumor se va a atenuar al transmitirse de agente a agente.
- Distribuciones. Dentro de los modelos de comportamiento, queremos que algunos parámetros tengan valores aleatorios pero dentro de una

distribución de probabilidad. Cada distribución posee un identificador único, un tipo de distribución y los parámetros de la misma (dependiendo de la distribución). Permitimos la creación de tres tipos de distribuciones de probabilidad: distribución normal, distribución uniforme y distribución de Poisson.

- Modelos psicológicos. El modelo psicológico es el modelo que determina como se comportará el agente sin tener en cuenta al resto de agentes. Son parámetros del estilo de máximo de acciones para comprar, porcentaje de bajada en el precio cuando no he podido comprar, etc... A estos parámetros se les puede asignar un valor constante o una distribución de probabilidad.
- Modelos sociales. Es similar al modelo psicológico pero con parámetros relacionados con el comportamiento social del agente. Son parámetros del estilo, número de conocidos o fiabilidad ante un rumor.
- Comportamientos. Por último está definido los distintos comportamientos de los agentes. Cada comportamiento debe tener los siguientes atributos:
 - Identificador único del comportamiento.
 - Clase que implementa el comportamiento escogido.
 - Modelo psicológico.
 - Modelo social.
 - Porcentaje de agentes. El tanto por ciento de agentes de cada contenedor de agentes que van a usar este comportamiento.
 - Subcomportamientos anidados. Se pueden anidar a cada comportamiento tantos comportamientos como se deseen, teniendo en cuenta que todos los comportamientos anidados van a usar los mismos modelos tanto psicológico como social.

9.13 Parámetros del sistema local

Una vez que el cliente de la aplicación inicia el sistema multiagente, tiene que proporcionar una serie de datos para el normal funcionamiento del sistema multiagente. Estos parámetros son:

- Numero de agentes. El número de agentes no es el número real de agentes que habrá en el sistema, sino que el sistema cogerá este indicador y los porcentajes que posee el fichero de configuración remoto

y generará el número de agentes correspondientes a cada tipo en el porcentaje indicado.

- Tiempo medio de espera. Dentro de la simulación el usuario tiene que indicar cuanto desea que esperen de media los agentes antes de tomar de nuevo decisiones.
- Tiempo de ciclo del sistema multiagente. Este parámetro indica lo rápido que pasa cada ciclo dentro del sistema de agentes.

9.14 Sistema clasificador

Un sistema clasificador (learning classifier system) es una técnica de aprendizaje máquina con relación directa con el aprendizaje por refuerzo y los algoritmos genéticos.

El sistema clasificador consta de dos partes, la primera es un conjunto de reglas codificadas por caracteres ternarios (0, 1, #).

La segunda parte es un algoritmo genético que se utiliza para poder hacer evolucionar las reglas y que los agentes aprendan de la interacción con el sistema. En vez de usar una función de aptitud, la utilidad de las reglas es decidida mediante técnicas de aprendizaje por refuerzo.

9.14.1 Codificación de las reglas

Como hemos comentado anteriormente, las reglas están formadas por cadenas de caracteres ternarios (0,1,#) donde # encaja tanto con 0 como con 1. Por ejemplo, la regla de la izquierda encaja con las dos reglas de la derecha.

00#100010100	000100010100
	001100010100

Las cadenas cuentan con 10 bits, donde cada bit significa lo siguiente

Bit	Condición
1	Periodo de 10 ciclos precio sube $\frac{1}{4}$
2	Periodo de 10 ciclos precio baja $\frac{1}{4}$
3	Periodo de 10 ciclos precio sube $\frac{1}{2}$

	Periodo de 10 ciclos precio baja $\frac{1}{2}$
4	Periodo de 30 ciclos precio sube $\frac{1}{4}$
5	Periodo de 30 ciclos precio baja $\frac{1}{4}$
6	Periodo de 30 ciclos precio sube $\frac{1}{2}$
7	Periodo de 30 ciclos precio baja $\frac{1}{2}$
8	Desde el principio de la jornada precio sube $\frac{1}{2}$
9	Desde el principio de la jornada precio baja $\frac{1}{2}$
10	Acción de compra si 1 de venta si 2

9.14.2 Acciones realizadas por las reglas

En cuanto a las acciones que realiza el sistema clasificador, depende de lo que llamaremos la “exactitud” de una regla. Con exactitud nos referimos a la cantidad de # que tiene, cuantos más # menos exacta es la regla. El bit 10 indica si se realiza una compra o una venta, la empresa elegida es única pues existe un clasificador por empresa, el precio siempre es el precio actual de la bolsa y la cantidad es siempre será de $\min + (\max - \min) * \text{exactitud}$. Donde max y min son los parametros de máximo compra de acciones y mínima compra de acciones respectivamente. Respecto a las ventas, siempre venderemos todas las que se poseen.

9.14.3 Asignación de beneficio a las reglas

Todas las compras y las ventas tienen que estar relacionadas. En un momento de la simulación, si hay una notificación de una venta, esta venta estará relacionada con todas las compras relacionadas previamente, y al revés. Si una hay una notificación de compra esta estará relacionada con todas las ventas que haya a continuación.

El beneficio se reparte siempre que hay una notificación de una venta, pues es cuando se hace efectivo el cobro del beneficio. En este momento se reparte entre todas las reglas relacionadas, siempre aportando la mitad del beneficio a la venta, y el resto se lo reparten las compras dependiendo del número de acciones que hayan aportado.

9.14.4 Algoritmo genético

- Función de aptitud

La función de aptitud viene dada por el beneficio asignado a las reglas y su similitud con las mismas. En un momento puntual de la simulación tendremos que las reglas del sistema clasificador han obtenido unos beneficios. Cuando apliquemos a esas reglas el algoritmo genético para hacerlas evolucionar, haremos que la aptitud sea el beneficio generado por cada regla multiplicado por su similitud, intentando dar siempre mayor valor de aptitud a las reglas que sean más exactas (que tengan menos símbolos #)

- Cruce

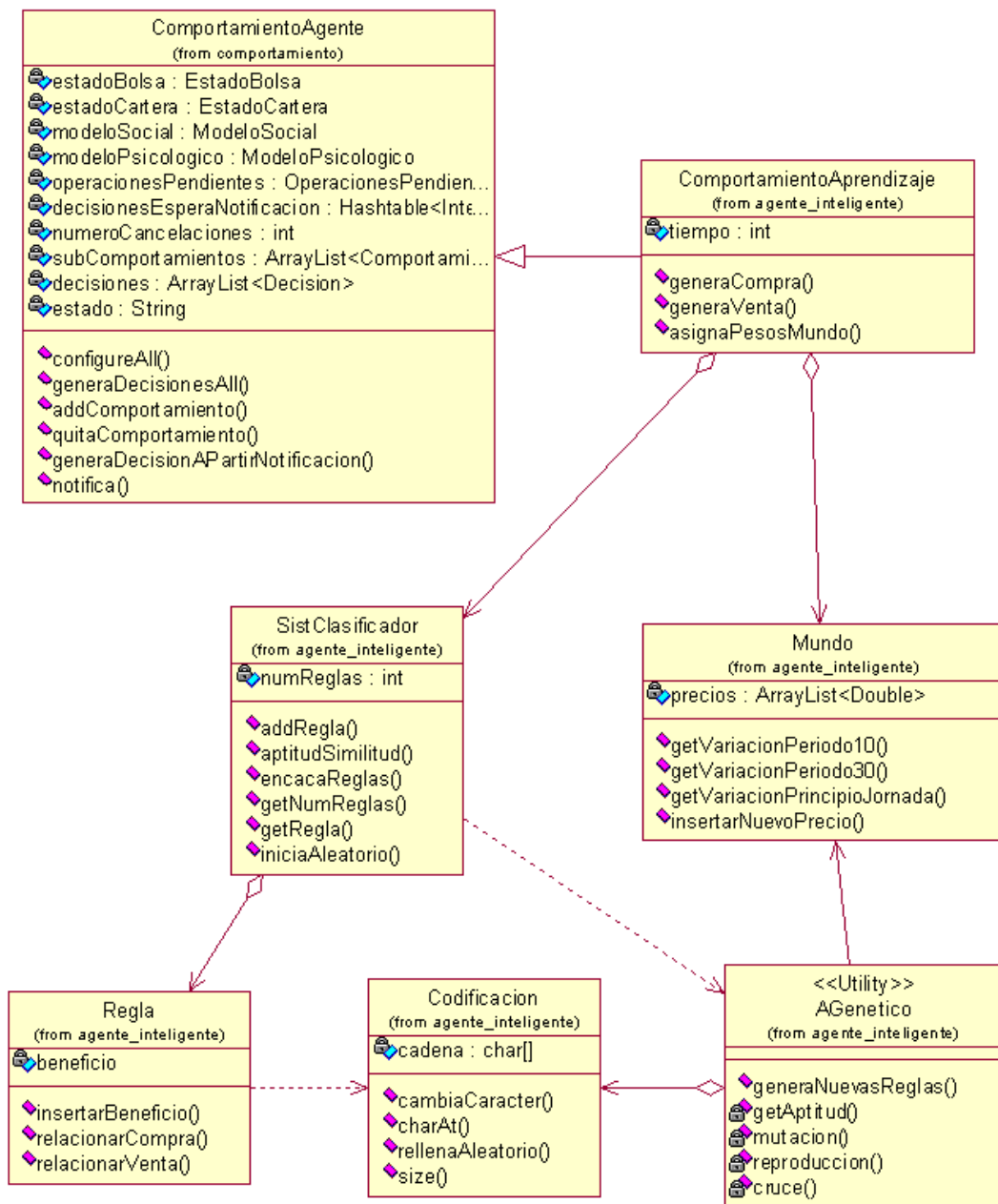
El cruce de dos cromosomas del sistema será el cruce normal de las cadenas binarias, es decir, se cogen los dos individuos seleccionados y un punto aleatorio de cruce. Después se generan dos nuevos individuos con los segmentos de las cadenas intercambiados.

- Mutación

Se generará una mutación puntual sobre la cadena, pudiendo mutar un solo carácter. Tiene la peculiaridad de que si el carácter a mutar es un 1 o un 0 y sale que debe haber mutación siempre va a cambiar a # y si es un # solo puede cambiar a 1 o a 0.

9.14.5 Diagrama de clases del sistema clasificador y el comportamiento.

Observamos como hemos estructurado el comportamiento mediante el sistema clasificador en el siguiente diagrama de clases. Primero tenemos que el comportamiento de aprendizaje extiende al comportamiento agente, y está compuesto por el sistema clasificador y el mundo que percibe. Cuando hay cambios en el mundo el comportamiento lo informa y eventualmente hace que el sistema clasificador genere nuevas reglas y evolucione a partir del mundo y de las reglas que contiene.



10. RIESGOS

Al ser un proyecto entre tres personas, los riesgos debidos a la falta de comunicación han sido inferiores que en proyectos de mayor envergadura. Pero los riesgos debidos a la indisponibilidad de alguno de los miembros son mucho mayores, pues tengamos en cuenta que si uno de los componentes del grupo esta indispuesto para la realización del mismo se trata del 33% de los componentes del grupo.

Los riesgos tecnológicos relacionados con los WebServices, al final no supusieron una complicación excesiva y pudimos solventarlos sin mayor problema.

10.1 Riesgos personales

- Falta de disponibilidad de alguno de los miembros

Riesgo dado debido a la falta de tiempo de los integrantes del grupo debido a prácticas, exámenes, etc.. de otras asignaturas.

Medidas para evitarlo: Aviso con antelación de la falta de disponibilidad para poder paliar, en la medida de lo posible, el retraso en la planificación.

Plan de contingencia: Realizar “horas extras” para recuperar el tiempo perdido.

- Falta de conocimientos en las tecnologías empleadas

Al emplear tecnologías que eran nuevas para nosotros, podría surgir el riesgo de perder excesivo tiempo en aprender a manejar las mismas.

Medidas para evitarlo: Dedicar tiempo al estudio de la tecnología antes de empezar a aplicarla al proyecto.

Plan de contingencia: Intentar que una persona que domine más la materia se cambie con la que no ha podido adquirir los conocimientos necesarios.

10.2 Riesgos tecnológicos

- Falta de suficiente eficiencia en los WebServices.

Al ser un sistema centralizado, y realizar la implementación de la conexión cliente-servidor mediante Web Services, se podría producir el problema de que no fuera lo suficientemente eficiente para poder manejar todas las peticiones que necesita nuestro sistema.

Medidas para evitarlo: Intentar optimizar al máximo todas las interacciones con los Web services. Cachear en la medida de lo posible, todos los datos obtenidos por el mismo.

Plan de contingencia: Informar al usuario de la limitación del sistema en lo relacionado con la eficiencia.

- Falta de escalabilidad en la herramienta de agentes.

En el sistema multiagente, al implementar cada agente mediante un hilo, podemos encontrarnos con el problema de que la herramienta no aguante demasiados agentes simultáneamente.

Medidas para evitarlo: Intentar minimizar los procesos que realizan los agentes.

Plan de contingencia: Probar la escalabilidad máxima en el número de agentes para poder informar al usuario de los límites del sistema

- Fallos en el CVS donde estaba alojado el proyecto.

Dependemos del CVS para guardar todo el código fuente del proyecto, si el servidor de CVS no esta disponible no podríamos tener actualizada la última versión del código.

Medidas para evitarlo: Guardar una copia de seguridad periódicamente una última versión del contenido del CVS.

Plan de contingencia: En caso de reiteradas caídas del servidor cambiar el sistema de versiones a SVN.

- Soporte a la concurrencia

Al ser un programa concurrente en el que además se ejecutan múltiples hilos de forma simultanea, tenemos que tener cuidado con la concurrencia y como actúan los agentes.

Medidas para evitarlo: Tener especial cuidado con la implementación del control concurrente del sistema.

Plan de contingencia: Serializar las operaciones para que no se realicen de forma concurrente.

10.3 Riesgos de estimación

- Fallos en la estimación del tiempo de desarrollo.

Al usar tecnologías desconocidas y al tener que usar librerías externas, no sabemos con exactitud el tiempo que nos va a llevar el desarrollar las distintas partes del proyecto. La estimación que hemos hecho antes de iniciar el

proyecto puede ser insuficiente una vez que tengamos que solucionar el problema concreto.

Medidas para evitarlo: Realizar revisiones periódicas en la planificación del proyecto.

Plan de contingencia: Hacer horas extras para recuperar los tiempos perdidos en la planificación, o intentar acortar los plazos para nuestras entregas internas.

- Retraso en los hitos planificados.

Por motivos ajenos al grupo, podría resultar que alguno de los hitos planificados deba ser retrasado, o pospuesto, a favor de otros que parecían menos urgentes pero que al final han resultado ser cruciales para otro subsistema.

Medidas para evitarlo: Planificar hitos realistas que sean alcanzables en el tiempo acordado.

Plan de contingencia: Reestructurar un hito complejo, que no se vaya a poder completar a tiempo, por dos o más hitos de menos complejidad y que se deben realizar en paralelo.

- Incorporación de nueva funcionalidad.

Puede que en el desarrollo del proyecto surja funcionalidad que no habíamos planificado, pero que sea necesaria para la integración de los distintos subsistemas. Esta nueva funcionalidad puede trastocar la planificación y retrasar todo el proyecto.

Medidas para evitarlo: Desarrollar los subsistemas al comienzo para poder determinar toda la funcionalidad.

Plan de contingencia: Desarrollar la nueva funcionalidad, y revisar la planificación para poder hacerlo.

10.4 Seguimiento de los riesgos

10.4.1 Riesgos producidos y planificados:

- Retrasos en los hitos planificados.

A pesar de nuestros esfuerzos por llevar el proyecto al día, nos ha sido imposible no retrasarnos un poco en diversas etapas del proyecto. Cuando surgían estos retrasos, posteriormente, teníamos que dedicar más horas al proyecto para recuperar el tiempo perdido.

- Fallos en el CVS donde estaba alojado el proyecto.

El CVS de BerliOS sufrió caídas a lo largo del proyecto. En concreto en 3 ocasiones no teníamos forma de subir los cambios al CVS. Afortunadamente estas caídas supusieron un máximo de un día de retraso en el desarrollo del proyecto.

- Falta de disponibilidad de alguno de los miembros

Debido a exámenes, prácticas y otras circunstancias personales, algunos de los miembros del equipo no pudieron realizar algunas de sus tareas a tiempo.

- Incorporación de nueva funcionalidad.

A medida que desarrollamos el proyecto tuvimos que añadir funcionalidad que no habíamos previsto, realizando una replanificación del proyecto.

10.4.2 Riesgos producidos y no planificados

- Nueva versión de la librería de los WebServices

En Mayo, salió la nueva versión de la librería para Web services de Apache, Axis2. Esta nueva versión traía muchas mejoras pero era totalmente incompatible con la versión antigua que era la que habíamos estado utilizando. Decidimos finalmente no actualizar la versión pues el esfuerzo extra que requería nos parecía excesivo en comparación con las mejoras que suponía para el proyecto.

11. PLANIFICACIÓN

11.1 Distribución del trabajo

En un grupo de tan solo tres personas se puede dividir el trabajo con mayor facilidad que en proyectos como el de Ingeniería del Software. También es más sencillo el ponerse de acuerdo para no solapar trabajo. Aunque muchas de las labores de integración de los subsistemas se hacían conjuntamente.

Al comienzo del proyecto todos estábamos dedicados a la documentación y al desarrollo de las interfaces del cliente y del servidor. Una vez finalizada la implementación de la funcionalidad básica tanto del cliente como del servidor, nos dividimos en subsistemas. Cliente, servidor, sistema de agentes, sistema de mensajes, sistema de eventos, sistema de bolsa y sistema de información. Lo primero que se finalizó fueron los subsistemas de mensajes, eventos y bolsa. Posteriormente se fueron implementando tanto las interfaces gráficas como los subsistemas de información y el comienzo del subsistema de contenedor de agentes.

Una vez finalizada la mayoría de los subsistemas y integrados, nos dedicamos a mejorar el sistema de agentes y el subsistema de interacción de los mismos.

11.2 Tareas realizadas

Las tareas a realizar se dividen en 4 grandes grupos.

S - tareas relacionadas con el servidor

C - tareas relacionadas con el cliente

A - tareas relacionadas con los agentes y

Sis - tareas relacionadas con el sistema global.

A su vez tenemos que una tarea puede ser de 4 tipos.

D - diseño

I - Implementación

P - pruebas

Doc - documentación

Esta notación la hemos especificado para facilitar la lectura de todas las tareas que vamos a definir y su posterior interpretación en el diagrama de Gantt.

DocSis1 - Estudio inicial del problema.

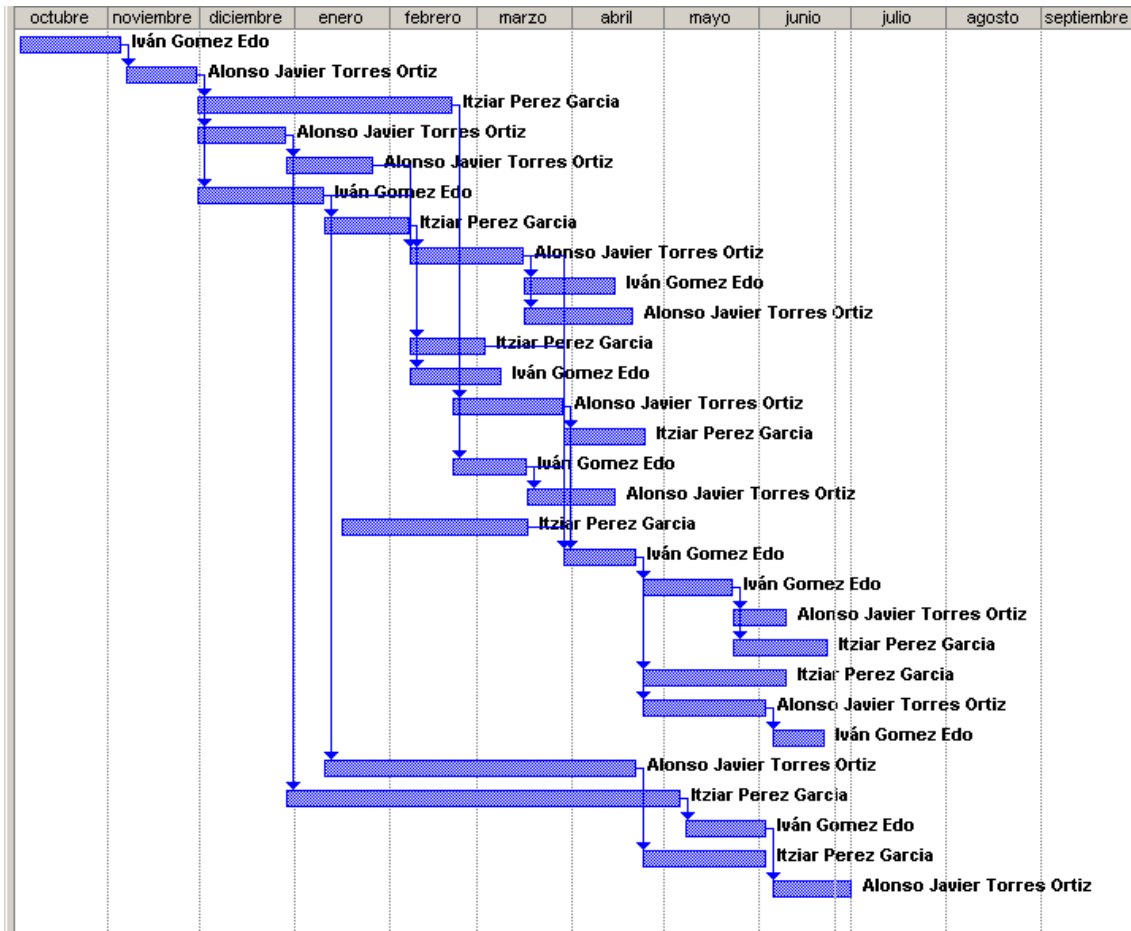
DSis1 - Diseño del sistema.

ISis1 - Definición de las interfaces remotas.
DS1 - Diseño de la arquitectura del servidor.
IS1 - Implementación de la funcionalidad básica del servidor.
DC1 - Diseño de la arquitectura del cliente.
IC1 - Implementación de la funcionalidad básica del cliente.
ISis2 - Interconexión cliente-servidor.
IS2 - Implementación de las reglas de la bolsa en el servidor.
IS3 - Pruebas con el cambio de precios y el cruce de operaciones.
IC2 - Implementación del sistema de información en el cliente.
IC3 - Creación de los ficheros y descarga de los mismos desde el cliente.
ISis2 - Implementación del sistema de eventos.
PSis1 - Pruebas del sistema de eventos.
ISis3 - Implementación del sistema de mensajes.
PSis2 - Pruebas del sistema de mensajes.
DA1 - Diseño de la arquitectura del sistema de agentes.
IA2 - Implementación inicial de los agentes.
IAC1 - Implementación del sistema de agentes del cliente.
IAC2 - Lectura de ficheros por parte del sistema de agentes.
IAS1 - Creación de los ficheros de configuración de agentes en el servidor.
IA3 - Mejoras en la interacción de los agentes.
IA4 - Mejoras en la estructura de los agentes.
IA5 - Implementación de los comportamientos iniciales.
DC2 - Diseño de las interfaces gráficas del cliente.
DS2 - Diseño de las interfaces gráficas del servidor.
PC - Pruebas finales del Cliente.
PS - Pruebas finales del Servidor.
PA - Pruebas finales de los Agentes.

11.3 Asignación de tareas

DocSis1	25 días	03/10/2005	04/11/2005	Iván Gomez Edo
DSis1	17 días	07/11/2005	29/11/2005 1	Alonso Javier Torres Ortiz
ISis1	59 días	30/11/2005	20/02/2006 2	Itziar Perez Garcia
DS1	21 días	30/11/2005	28/12/2005 2	Alonso Javier Torres Ortiz
IS1	20 días	29/12/2005	25/01/2006 4	Alonso Javier Torres Ortiz
DC1	29 días	30/11/2005	09/01/2006 2	Iván Gomez Edo
IC1	20 días	10/01/2006	06/02/2006 6	Itziar Perez Garcia
ISis2	27 días	07/02/2006	15/03/2006 5;7;6	Alonso Javier Torres Ortiz
IS2	22 días	16/03/2006	14/04/2006 8	Iván Gomez Edo
IS3	26 días	16/03/2006	20/04/2006 8	Alonso Javier Torres Ortiz
IC2	19 días	07/02/2006	03/03/2006 7	Itziar Perez Garcia
IC3	22 días	07/02/2006	08/03/2006 7	Iván Gomez Edo
ISis2	26 días	21/02/2006	28/03/2006 3	Alonso Javier Torres Ortiz
PSis1	19 días	29/03/2006	24/04/2006 13	Itziar Perez Garcia
ISis3	18 días	21/02/2006	16/03/2006 3	Iván Gomez Edo
PSis2	21 días	17/03/2006	14/04/2006 15	Alonso Javier Torres Ortiz
DA1	45 días	16/01/2006	17/03/2006	Itziar Perez Garcia
IA2	18 días	29/03/2006	21/04/2006 8;11;13;15;17	Iván Gomez Edo
IAC1	21 días	24/04/2006	22/05/2006 18	Iván Gomez Edo
IA3	14 días	23/05/2006	09/06/2006 19	Alonso Javier Torres Ortiz
IA4	23 días	23/05/2006	22/06/2006 19	Itziar Perez Garcia
IAC2	35 días	24/04/2006	09/06/2006 18	Itziar Perez Garcia
IAS1	30 días	24/04/2006	02/06/2006 18	Alonso Javier Torres Ortiz
IA5	13 días	05/06/2006	21/06/2006 23	Iván Gomez Edo
DC2	74 días	10/01/2006	21/04/2006 6	Alonso Javier Torres Ortiz
DS2	92 días	29/12/2005	05/05/2006 4	Itziar Perez Garcia
PC	20 días	08/05/2006	02/06/2006 26	Iván Gomez Edo
PS	30 días	24/04/2006	02/06/2006 25	Itziar Perez Garcia
PA	20 días	05/06/2006	30/06/2006 27	Alonso Javier Torres Ortiz

11.4 Diagrama de GANTT



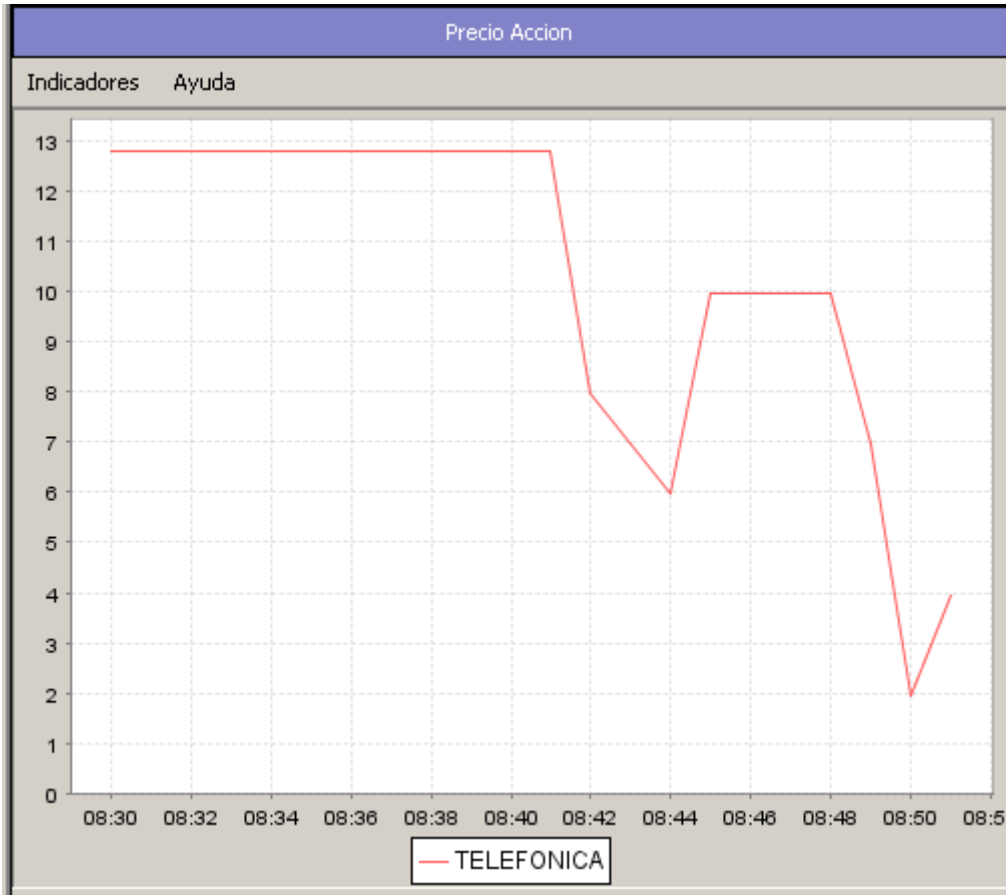
12. Pruebas

12.1. Prueba 1: Cómo influir sobre una empresa

Supongamos que estamos en un sistema en el cual tengo dinero casi ilimitado como sucede con algunos de los grandes broker de la bolsa los demás agentes interactúan con las demás empresas entonces vamos a ser malévolos y bajamos los precios de Telefónica empresa de mayor volumen en el IBEX 35 para ello debemos comprar acciones por valores debajo del actual a grandes cantidades la gente se asusta y empieza a vender lo que provocará que las acciones bajen, vemos peticiones de compra a la baja.

Operaciones pendientes <input checked="" type="radio"/>		Mi cartera de acciones <input type="radio"/>			
	IDOperacion	Tipo	Empresa	Cantidad	Precio
✘	87	COMPRA	TELEFONICA	100	8.0
✘	110	COMPRA	TELEFONICA	100	6.0
✘	155	COMPRA	TELEFONICA	100	4.0

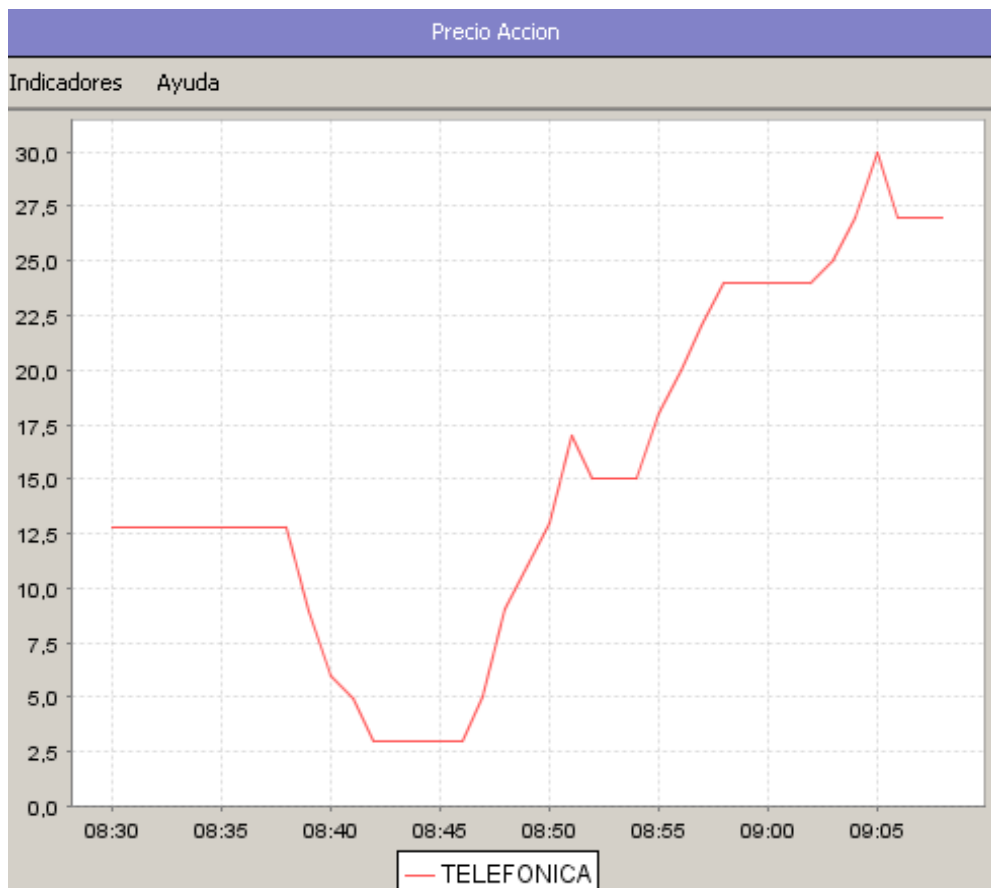
Observamos en el gráfico como va descendiendo rápidamente el valor de las acciones.



Ahora volvemos a introducir compras a valores mayores pero en menor cantidad para subir de nuevo la empresa rápidamente.

Operaciones pendientes <input checked="" type="radio"/>		Mi cartera de acciones <input type="radio"/>			
IDOperacion	Tipo	Empresa	Cantidad	Precio	
✗ 5	COMPRA	TELEFONICA	200	17.0	
✗ 6	COMPRA	TELEFONICA	200	15.0	
✗ 7	COMPRA	TELEFONICA	200	13.0	
✗ 8	COMPRA	TELEFONICA	200	11.0	
✗ 9	COMPRA	TELEFONICA	200	9.0	
✗ 10	COMPRA	TELEFONICA	200	6.0	
✗ 11	COMPRA	TELEFONICA	200	5.0	

ahora vemos como todas las peticiones se procesan eventualmente y el precio de la acción comienza a subir con lo que provocará una media mucho más baja y un aumento de acciones que producirá un aumento en los beneficios.



Vemos como el precio medio nos queda mucho mejor que el actual que ronda los 28. Beneficio-> $4,5 \cdot 2000 = 9000$ euros.

Operaciones pendientes <input type="radio"/> Mi cartera de acciones <input checked="" type="radio"/>		
Empresa	Numero Acciones	Precios Medio por Accion
TELEFONICA	2000	23.5

12.2. Prueba2: Interacción entre clientes de distintas máquinas

Vamos a comprobar que dos clientes de distintas máquinas introducen una orden de compra y otro una orden de venta y veamos que se cruzan.

Primero el Agente Alonso introduce una venta. Por que ya tenia acciones de telefónica.

Operaciones pendientes <input type="radio"/> Mi cartera de acciones <input checked="" type="radio"/>		
Empresa	Numero Acciones	Precios Medio por Accion
TELEFONICA	150	14.0

Operaciones pendientes <input checked="" type="radio"/> Mi cartera de acciones <input type="radio"/>					
	IDOperacion	Tipo	Empresa	Cantidad	Precio
✗	4	VENTA	TELEFONICA	150	16.0

Ahora desde el Agente Iván ponemos una orden de compra a 16

Operaciones pendientes <input checked="" type="radio"/> Mi cartera de acciones <input type="radio"/>					
	IDOperacion	Tipo	Empresa	Cantidad	Precio
✗	6	COMPRA	TELEFONICA	150	16.0

Después de un rato esperando se produce el cruce de acciones con lo que las acciones pasan de la cartera del Agente alonso a la cartera del Agente Iván.

Cartera Alonso

Operaciones pendientes <input type="radio"/> Mi cartera de acciones <input checked="" type="radio"/>		
Empresa	Numero Acciones	Precios Medio por Accion

Cartera Iván

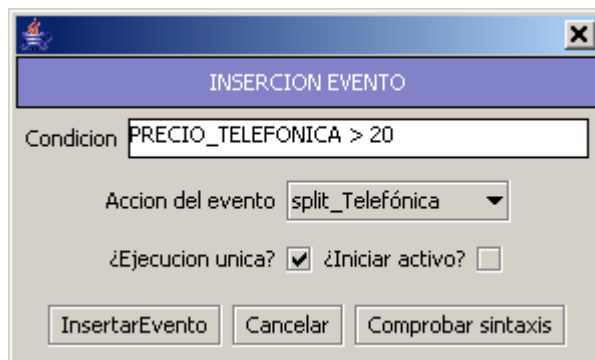
Operaciones pendientes <input type="radio"/> Mi cartera de acciones <input checked="" type="radio"/>		
Empresa	Numero Acciones	Precios Medio por Accion
TELEFONICA	150	16.0

Con lo que alonso ha ganado 2 euros por acción de las 150 que tenia = 300 euros de beneficio.

12.3. Prueba 3: Aviso de ampliación de capital (de tipo split) sobre un valor

El administrador para evitar el aumento masivo del precio de un valor quiere poner un aviso de split en este caso de telefónica para ello introducirá un evento avisando a los clientes de este hecho.

Introducción del evento:



INSERCIÓN EVENTO

Condicion: PRECIO_TELEFONICA > 20

Accion del evento: split_Telefónica

¿Ejecucion unica? ¿Iniciar activo?

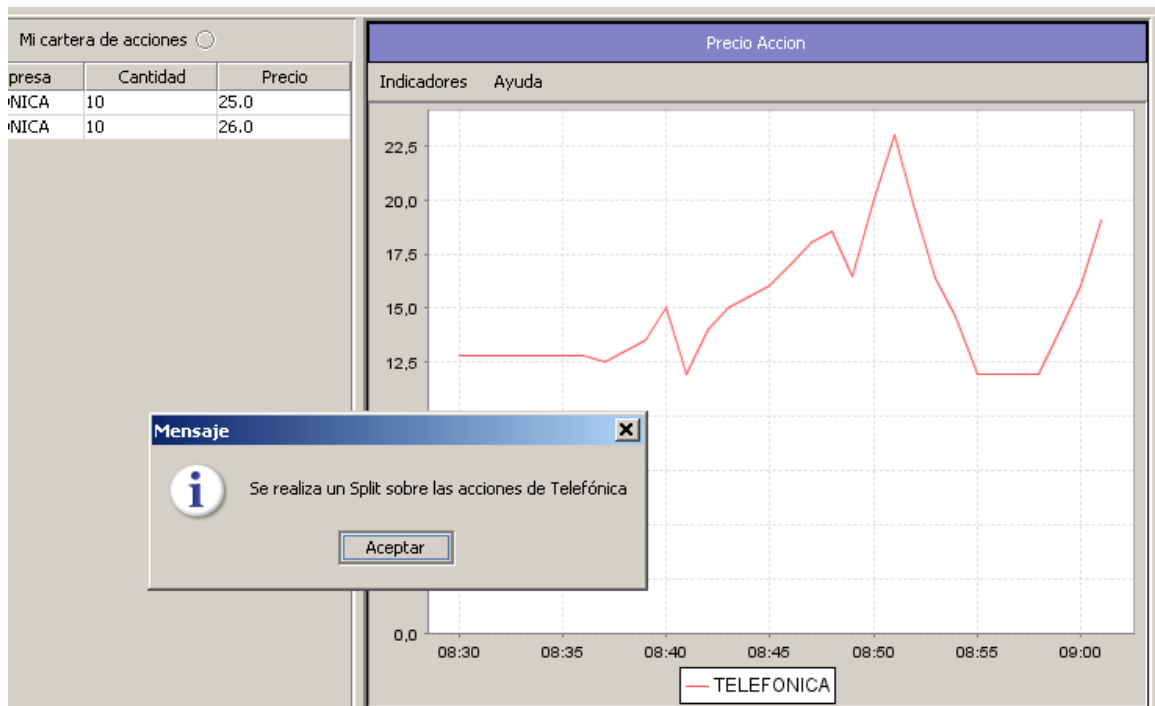
InsertarEvento Cancelar Comprobar sintaxis

Cuando el precio supere los 20 euros se informaran a los clientes vemos como queda introducido el evento.



Control de Eventos		
Condicion	Accion	Activo?
PRECIO_TELEFONICA > 20	SPLIT_Telefónica	<input checked="" type="checkbox"/>

Los usuarios estarán operando normalmente hasta que ocurran las condiciones del evento en este caso se activará el aviso emergente



12.4. Prueba 4: actuación de 300 agentes y 2 clientes en el sistema

Vamos a iniciar dos clientes cada uno con 150 agentes y vamos a dejarles que interactúen con el sistema durante un periodo de tiempo, veamos los agentes generados desde:

Cliente Iván

Agentes en el sistema					Notificaciones de los agentes
IDAgente	Estado Actual	Comportamiento	Ganancias		
<input type="checkbox"/> ivan#Agente1	Genera compra	Aleatorio	-6.641,19		ivan#Agente13: Operacion efectuada: 286(886,16.81)
<input type="checkbox"/> ivan#Agente2	Genera compra	Aleatorio	0		ivan#Agente113: Operacion efectuada: 290(228,9.68)
<input type="checkbox"/> ivan#Agente3	Genera compra	Aleatorio	0		ivan#Agente116: Operacion efectuada: 221(371,16.46)
<input type="checkbox"/> ivan#Agente4	Genera compra	Aleatorio	0		ivan#Agente37: Operacion efectuada: 340(558,12.61)
<input type="checkbox"/> ivan#Agente5	Genera compra	Aleatorio	-10.398,6		ivan#Agente108: Operacion efectuada: 285(965,35.15)
<input type="checkbox"/> ivan#Agente6	Genera compra	Aleatorio	0		ivan#Agente31: Operacion efectuada: 320(211,19.86)
<input type="checkbox"/> ivan#Agente7	Genera compra	Aleatorio	0		ivan#Agente69: Operacion efectuada: 333(600,1.57)
<input type="checkbox"/> ivan#Agente8	Genera compra	Aleatorio	0		ivan#Agente29: Operacion efectuada: 304(876,28.93)
<input type="checkbox"/> ivan#Agente9	Genera compra	Aleatorio	0		ivan#Agente70: Operacion efectuada: 295(467,24.13)
<input type="checkbox"/> ivan#Agente10	Genera compra	Aleatorio	0		ivan#Agente67: Operacion efectuada: 332(975,13.66)
<input type="checkbox"/> ivan#Agente11	Genera compra	Aleatorio	-7.221,9		ivan#Agente118: Operacion efectuada: 214(259,11.19)
<input type="checkbox"/> ivan#Agente12	Genera compra	Aleatorio	0		ivan#Agente125: Operacion efectuada: 205(594,118.2)
<input type="checkbox"/> ivan#Agente13	Genera compra	Aleatorio	0		ivan#Agente26: Operacion efectuada: 275(281,22.71)
<input type="checkbox"/> ivan#Agente14	Genera compra	Aleatorio	0		ivan#Agente28: Operacion efectuada: 300(313,22.71)
<input type="checkbox"/> ivan#Agente15	Genera compra	Aleatorio	0		
<input type="checkbox"/> ivan#Agente16	Genera compra	Aleatorio	0		
<input type="checkbox"/> ivan#Agente17	Genera compra	Aleatorio	0		

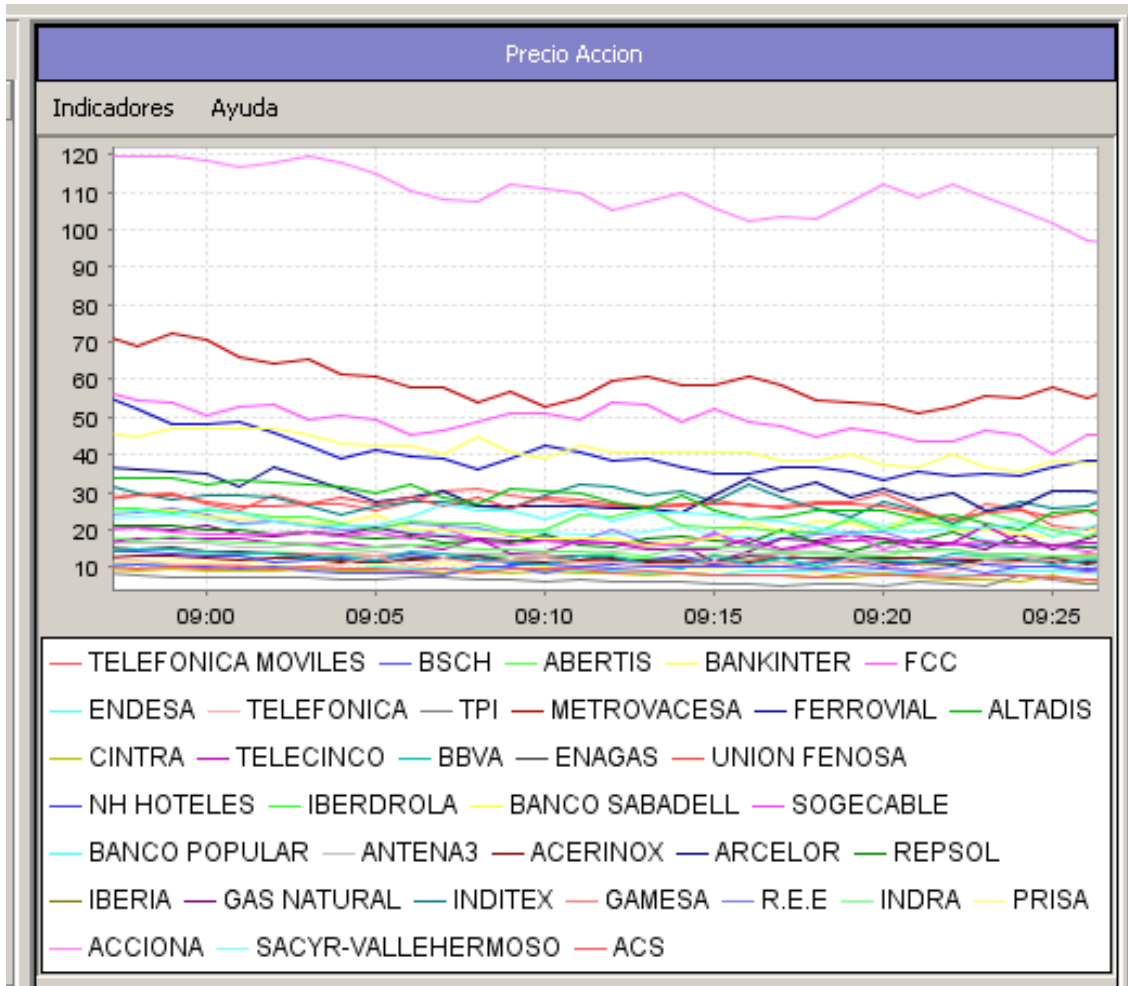
Acciones de los agentes
ivan#Agente146: Espero recepcion mensaje (PERFORMATIVE,REQUEST)(PROTOCOL,REQUEST)

Cliente Alonso

Agentes en el sistema					Notificaciones de los agentes
IDAgente	Estado Actual	Comportamiento	Ganancias		
<input type="checkbox"/> alonso#Agente1	Genera compra	Aleatorio	-22.558,9		alonso#Agente33: Operacion efectuada: 360(462,26.33)
<input type="checkbox"/> alonso#Agente2	Genera cancelacion	Aleatorio	0		alonso#Agente33: Operacion efectuada: 172(411,23.48)
<input type="checkbox"/> alonso#Agente3	Genera compra	Aleatorio	-37.253,74		alonso#Agente74: Operacion efectuada: 350(103,23.63)
<input type="checkbox"/> alonso#Agente4	Genera compra	Aleatorio	0		alonso#Agente100: Operacion efectuada: 358(622,18.72)
<input type="checkbox"/> alonso#Agente5	Genera compra	Aleatorio	-9.100,65		alonso#Agente51: Operacion efectuada: 344(185,10.18)
<input type="checkbox"/> alonso#Agente6	Genera compra	Aleatorio	-38.621,16		alonso#Agente52: Operacion efectuada: 346(760,10.18)
<input type="checkbox"/> alonso#Agente7	Genera compra	Aleatorio	-1.201,92		alonso#Agente118: Operacion efectuada: 260(860,12.31)
<input type="checkbox"/> alonso#Agente8	Genera compra	Aleatorio	-44.234,83		alonso#Agente62: Operacion efectuada: 96(642,36.4)
<input type="checkbox"/> alonso#Agente9	Genera compra	Aleatorio	-21.873,75		alonso#Agente87: Operacion efectuada: 426(674,18.91)
<input type="checkbox"/> alonso#Agente10	Genera compra	Aleatorio	-29.174,76		alonso#Agente39: Operacion efectuada: 161(149,28.83)
<input type="checkbox"/> alonso#Agente11	Genera compra	Aleatorio	-13.986,14		alonso#Agente108: Operacion efectuada: 386(926,22.53)
<input type="checkbox"/> alonso#Agente12	Genera compra	Aleatorio	-13.919,4		alonso#Agente102: Operacion efectuada: 336(295,14.41)
<input type="checkbox"/> alonso#Agente13	Genera compra	Aleatorio	-7.196,51		alonso#Agente87: Operacion efectuada: 423(852,10.79)
<input type="checkbox"/> alonso#Agente14	Genera compra	Aleatorio	-19.946,97		alonso#Agente90: Operacion efectuada: 179(366,117.75)
<input type="checkbox"/> alonso#Agente15	Genera compra	Aleatorio	-7.178,04		
<input type="checkbox"/> alonso#Agente16	Genera compra	Aleatorio	-14.275,38		
<input type="checkbox"/> alonso#Agente17	Genera compra	Aleatorio	-5.860,4		
<input type="checkbox"/> alonso#Agente18	Genera compra	Aleatorio	-36,966		
<input type="checkbox"/> alonso#Agente19	Genera compra	Aleatorio	-59.669,5		

Acciones de los agentes
alonso#Agente55: Espero hasta el ciclo 147
alonso#Agente41: Introducir operacion COMPRA-alonso#Agente41(ENI
alonso#Agente41: Espero hasta el ciclo 147
alonso#Agente57: Introducir operacion COMPRA-alonso#Agente57(ENI
alonso#Agente45: Introducir operacion COMPRA-alonso#Agente45(ENI

Después de un breve periodo en el que los agentes realizan sus operaciones de compra y venta el sistema queda con la gráfica de todas las acciones de la siguiente forma, con lo que vemos que el sistema aguanta y fluctúa correctamente independientemente de las órdenes introducidas.



13. Conclusiones

Hemos realizado un sistema de simulación bursátil, está es la principal diferencia con la mayoría de sistemas que hay actualmente en el mercado (bloomberg, visualchart,...) ya que no nos limitamos a mostrar la información de los valores en tiempo real sino que intentamos simular a partir de unos valores reales.

Para poder realizar una simulación del mercado bursátil mucho más aproximada y que este proyecto demostrara su utilidad real, se necesitaría mucho más tiempo de desarrollo y un equipo de mayor envergadura ya que deberíamos desarrollar una inteligencia muy avanzada para que los agentes pudiesen simular personas físicas que interactúan con la bolsa.

A pesar de esta dificultad, nuestro sistema simula obteniendo datos bastante aproximados a lo que sería de esperar, y aunque hay aprendizaje en alguno de los agentes implementados no hay aprendizaje social.

Creemos que este proyecto es muy didáctico pues abarca gran parte de las áreas que estudiamos dentro de la carrera (inteligencia artificial, programación distribuida, ingeniería del software, etc...) además nos ha brindado la oportunidad de conocer algunas herramientas nuevas como TOMCAT, JDK 1.5, Eclipse, etc... Además hemos aprendido una forma de comunicación entre máquinas remotas para transmitir información que está en auge, los Web Services.

Esta idea desarrollada durante 9 meses puede ser una herramienta de aprendizaje para los profesionales del campo de inversión ya que es posible aprender como funcionan los valores, la psicología de los agentes etc..

Se podrían realizar bastantes mejoras en el sistema implementado como por ejemplo la ya comentada mejora en la inteligencia de los agentes, añadir herramientas e indicadores de análisis de valores, añadir más valores, distintos mercados distintos tipos de acciones como pueden ser warrants, futuros y opciones.

Pero el tiempo del proyecto es limitado y somos 3 personas con lo cual nos centramos en el IBEX 35 que son los valores más importantes del mercado continuo madrileño.

APENDICE A: GLOSARIO

Ampliación de capital: El precio de las nuevas acciones depende de cómo se realice la ampliación, que puede ser:

- Ampliación a la par: el precio de las nuevas acciones coincide con el valor nominal de la acción.
- Ampliación con prima de emisión: el precio de las nuevas acciones es el valor nominal más una cantidad, que es la prima de emisión.
- Emisión de acciones liberadas a los accionistas actuales. Los accionistas no tienen que hacer ningún desembolso.

Los accionistas actuales tienen un derecho de suscripción preferente que da preferencia a la hora de comprar nuevas acciones en caso de ampliación de capital, lo que les permite comprar en proporción al nº de acciones de la compañía que posean actualmente.

Desequilibrio: Diferencia entre el volumen de oferta y de demanda.

Histórico: Conjunto de valores que ha tomado una acción a lo largo del tiempo.

Mercado primario: Mercado de emisión de títulos es decir es el mercado de los títulos cuando salen por primera vez a la venta. Se complementa con el mercado secundario.

Mercado secundario: Mercado de intercambio de títulos previamente emitidos y que ya figuran en posesión de los inversores. Por ejemplo: las bolsas de valores.

Oferta pública de adquisición (OPA): Operación mediante la cual una persona física o jurídica ofrece a los accionistas de una compañía cotizada la adquisición de sus acciones, a un determinado precio (normalmente por encima del valor actual de cotización), con el objetivo de obtener una participación significativa en el capital de esa sociedad.

Oferta pública de venta (OPV): Se realiza con el fin de vender una parte o la totalidad del capital social de una compañía al público en general; permite diseminar la propiedad, para garantizar la liquidez antes de su admisión a cotización, o incrementarla si ya cotiza. Se trata de un mecanismo utilizado en los siguientes casos:

- Privatizaciones o salidas a bolsa de empresas públicas.
- Salida a bolsa de empresas privadas. Cuando una empresa decide salir a cotizar, es frecuente que, con carácter previo, sea necesario vender un porcentaje de su capital al público para alcanzar la difusión requerida.

Prima de emisión: Diferencia entre el valor nominal de un título y el precio por el que se emite. En una emisión de acciones, por ejemplo, esta prima es un sobreprecio que abonan los nuevos accionistas.

Split: Consiste en dividir el valor nominal de las acciones de una empresa en una proporción determinada, y multiplicar el número de acciones en esa misma proporción. La finalidad de esta operación es proporcionar mayor liquidez a las acciones de la empresa y mejorar sus volúmenes de títulos contratados, mediante la reducción del valor de las acciones en el mercado. Normalmente implica un aumento en el precio de las acciones a corto plazo.

Esta operación no supone ningún desembolso por parte del accionista; en realidad no tiene ningún efecto económico-financiero pero posiblemente sí psicológico, al reducirse el precio de las acciones en la proporción establecida. Las proporciones más usadas son: 2 por 1, 3 por 1, 3 por 2, 4 por 3, 5 por 2 y 5 por 4.

Tick: Es la variación mínima exigida en el precio de un valor negociado en el SIBE. Depende del precio del valor de que se trate: si la acción vale menos de 50 euros, la cotización variará como mínimo 0,01 euros; si supera los 50, el tick será de 0,05 euros.

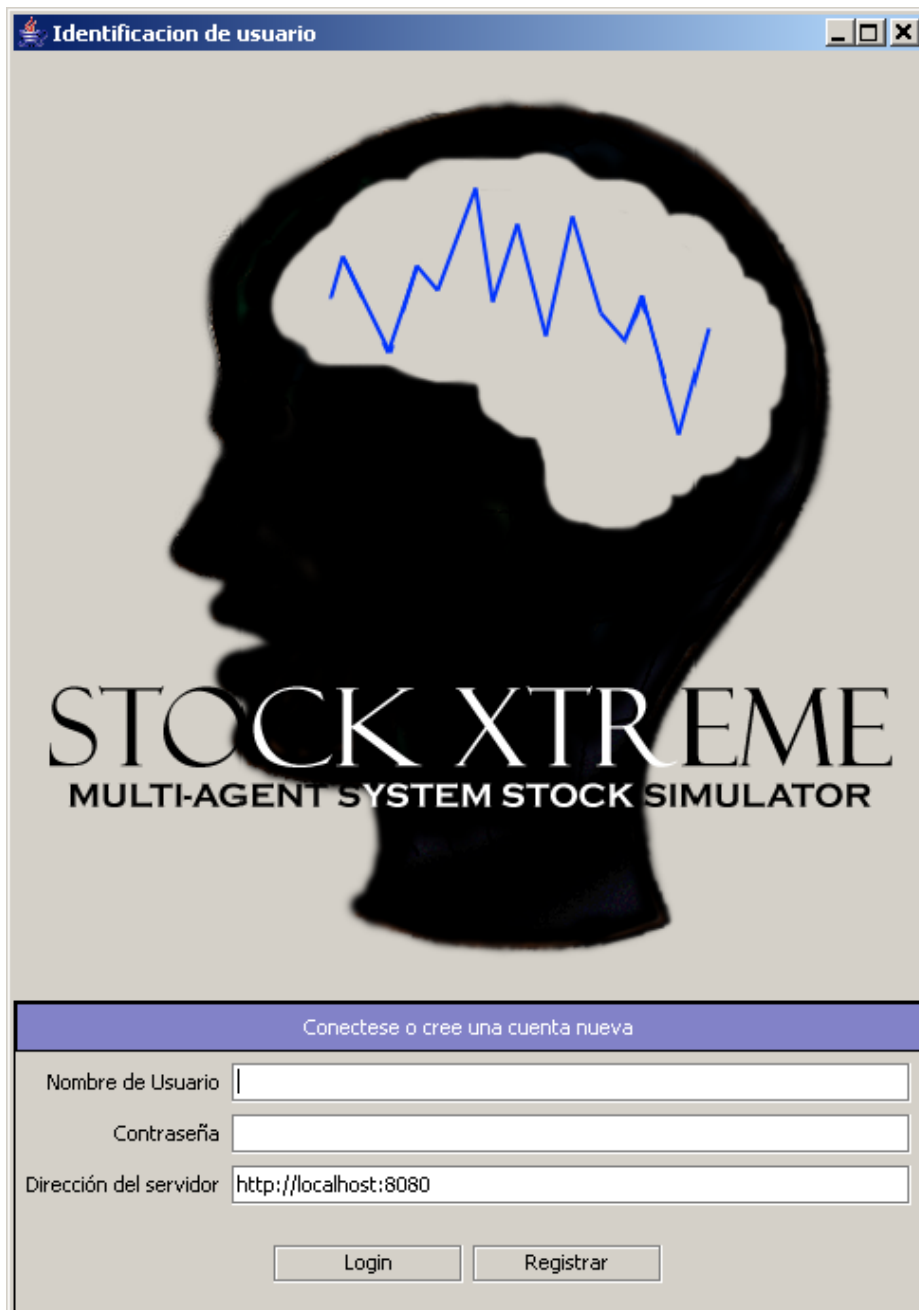
Valor nominal: Resultado de dividir el capital social de la empresa entre el número de acciones.

Volumen de contratación: Cantidad de valores o acciones contratados en un periodo de tiempo.

APENDICE B: MANUAL DE USUARIO

Para facilitar al cliente el uso del software vamos a explicar paso a paso todos los posibles usos del mismo:

Primer contacto pantalla de registro, al iniciar el programa, aparece la siguiente ventana.



The screenshot shows a window titled "Identificación de usuario" with a standard Windows title bar. The main content area features a large graphic of a human head in profile, facing left, with a brain inside. A blue line graph, representing stock market fluctuations, is overlaid on the brain. Below the graphic, the text "STOCK XTREME" is written in a large, serif font, and "MULTI-AGENT SYSTEM STOCK SIMULATOR" is written in a smaller, sans-serif font below it. At the bottom of the window, there is a purple header bar with the text "Conectese o cree una cuenta nueva". Below this bar are three input fields: "Nombre de Usuario" (empty), "Contraseña" (empty), and "Dirección del servidor" (containing "http://localhost:8080"). At the very bottom, there are two buttons: "Login" and "Registrar".

Si ya se había registrado previamente rellene los datos pulse login pero en el caso que aun no esté registrado rellene los datos y pulse en el botón Registrar.

Ejemplo de registro

Conectese o cree una cuenta nueva

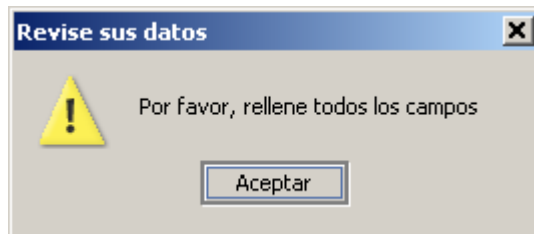
Nombre de Usuario: usuarioNuevo

Contraseña: *****

Dirección del servidor: http://localhost:8080

Login Registrar

En el caso que no introduzcamos alguno de los datos Stock Xtreme nos avisa de dicho fallo.



Por ultimo la dirección del servidor es la que nos proporciona por defecto Stock Xtreme no debe modificarse.

Una vez que ya estamos registrados entramos a la interfaz gráfica principal del usuario que tiene este aspecto.

Stock Xtreme

Principal Información Agentes

Operaciones pendientes Mi cartera de acciones

IDOperacion	Tipo	Empresa	Cantidad	Precio
-------------	------	---------	----------	--------

Indicadores Ayuda

500
480
460
440
420
400
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0

09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00

TELEFONICA MOVILES

Insertar una Operación

Empresa: TELEFONICA MOVILES

Tipo Operación: Compra

Numero de acciones:

Precio de compra:

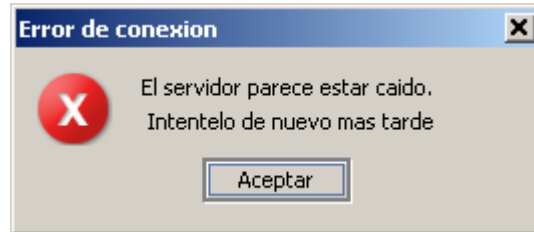
Insertar Operación Limpiar Campos

Empresas

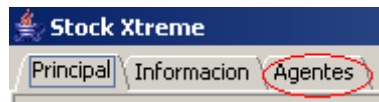
Empresa	Precio Actual	Ver en grafico
TELEFONICA MOVILES	10.8	<input checked="" type="checkbox"/>
BSCH	11.0	<input type="checkbox"/>
ABERTIS	18.17	<input type="checkbox"/>
BANKINTER	48.28	<input type="checkbox"/>
FCC	59.05	<input type="checkbox"/>
ENDESA	26.95	<input type="checkbox"/>
TELEFONICA	12.81	<input type="checkbox"/>
TPI	8.47	<input type="checkbox"/>

TELEFONICA MOVILES = 0

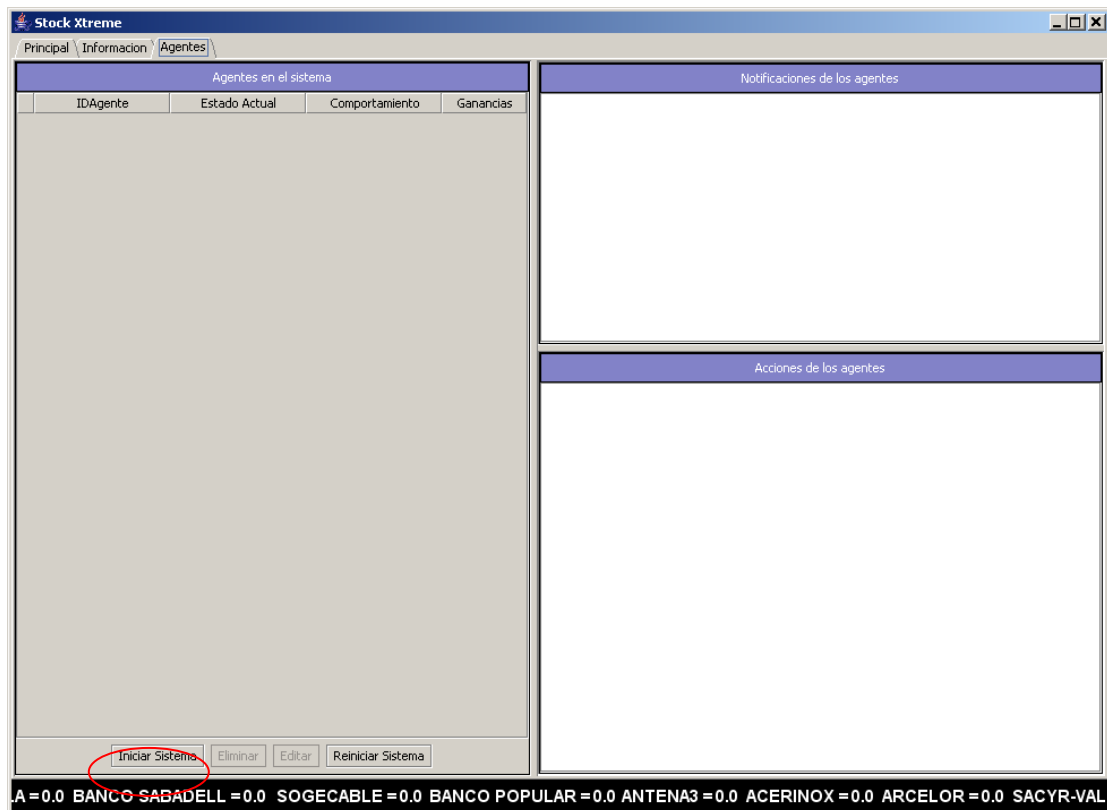
Si por algún motivo no este el servidor en funcionamiento o la jornada no se haya iniciado Stock Xtreme nos avisara de ello.



Una vez iniciada la aplicación correctamente antes de empezar ha realizar algún movimiento en la bolsa que esta funcionando en tiempo real debemos iniciar a los agentes automáticos que interactuaran con usted en la bolsa para ello pulsamos la pestaña Agentes



Nos aparece la ventana de agentes con lo que debemos pulsar el botón de "iniciar sistema" como aparece en el siguiente dibujo.



Con lo que nos aparecerá la pantalla de configuración de los Agentes que seleccionaremos los parámetros requeridos.

Una vez que iniciamos los Agentes nos aparecerá con su comportamiento en una ventana

Agentes en el sistema				
	IDAgente	Estado Actual	Comportamiento	Ganancias
<input type="checkbox"/>	ivan#Agente1	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente2	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente3	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente4	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente5	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente6	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente7	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente8	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente9	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente10	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente11	Genera compra	Aleatorio	0
<input type="checkbox"/>	ivan#Agente12	Genera compra	Aleatorio	0

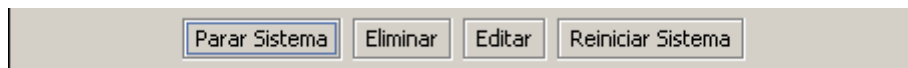
Veamos en la ventana también las notificaciones que se producen en los agentes que ha iniciado.

Notificaciones de los agentes
ivan#Agente56: Operacion efectuada: 33(178,18.88)
ivan#Agente51: Operacion efectuada: 46(277,16.06)
ivan#Agente36: Operacion efectuada: 72(734,12.91)
ivan#Agente23: Operacion efectuada: 31(417,33.75)
ivan#Agente21: Operacion efectuada: 18(93,23.36)
ivan#Agente12: Operacion efectuada: 32(478,1.92)
ivan#Agente13: Operacion efectuada: 38(73,1.92)
ivan#Agente84: Operacion efectuada: 12(934,22.01)
ivan#Agente15: Operacion efectuada: 56(763,31.98)
ivan#Agente62: Operacion efectuada: 64(163,14.96)
ivan#Agente43: Operacion efectuada: 20(193,24.08)
ivan#Agente32: Operacion efectuada: 66(986,14.51)
ivan#Agente17: Operacion efectuada: 68(827,11.89)
ivan#Agente35: Operacion efectuada: 82(565,115.4)
ivan#Agente67: Operacion efectuada: 76(690,30.43)

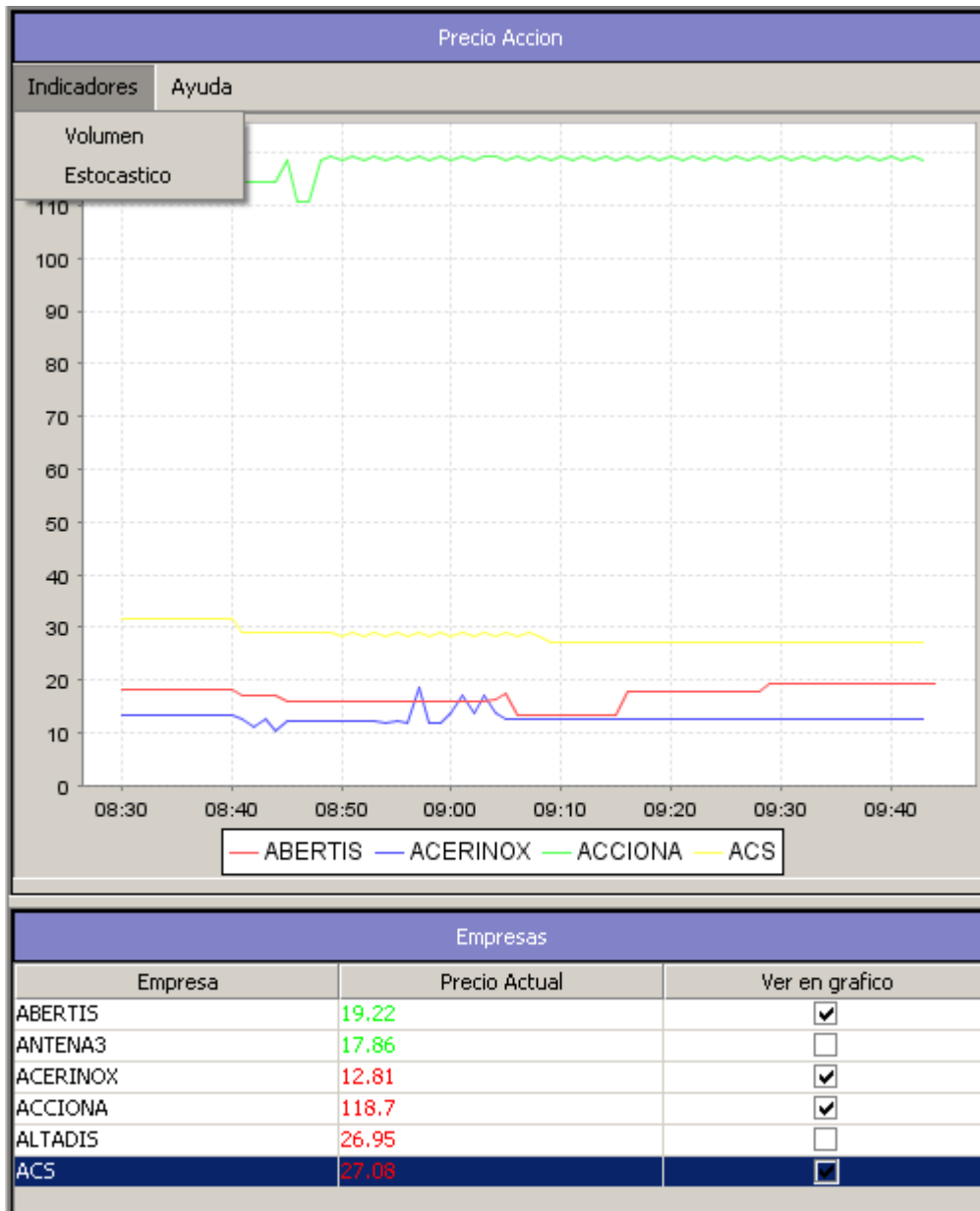
Vemos en la tabla inferior a la anterior las acciones que producen todos los agentes



Por ultimo tendremos la posibilidad de eliminar agentes o realizar un seguimiento exhaustivo de algún agente en concreto.



Ahora volvemos a la pestaña inicial que es donde vamos a trabajar con las acciones y valores disponibles tenemos la parte de la derecha los gráficos y precio de los valores disponibles, mediante los ticks podremos ver en el gráfico como fluctúan los valores intradia de las empresas deseadas también tenemos la opción de utilizar a modo ayuda los indicadores sobre los valores de empresa, se proporciona una ayuda en tiempo de ejecución para facilitar su uso



Veamos el funcionamiento de las herramientas, existen 2:

- 1.- Volumen: muestra el volumen de ventas de una acción en un determinado tiempo elegido por el usuario.
- 2.- Estocástico: muestra el estado de un valor si esta sobre comprado o sobre vendido para obtener el futuro comportamiento del mismo.

Supongamos que hemos ejecutado ambas herramientas debemos elegir la fecha mediante el siguiente cuadro emergente

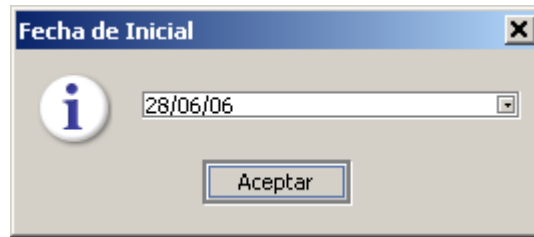


Gráfico de las herramientas:



Para introducir una orden de compra o de venta utilizamos la parte habilitada para introducir el número de acciones y el precio de la acción para introducir el precio de la acción tenemos dos opciones que el sistema nos ponga el precio actual o que elijamos el que más nos conviene.

Insertar una Operación

Empresa: BANCO POPULAR

Tipo Operación: Compra

Numero de acciones: 100

Precio de compra: 11.18

Insertar Operación Limpiar Campos

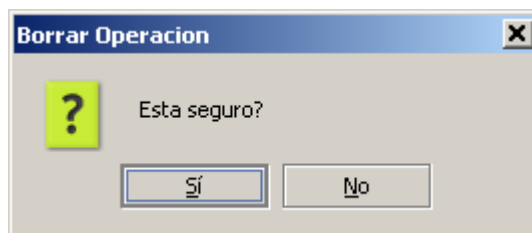
Y a continuación pulsamos Insertar Operación por si nos equivocamos tenemos la opción de limpiar los campos y volver a introducir.

Una vez que nuestra orden ha sido introducida se mostrará en la siguiente tabla hasta que haya sido procesado.

Operaciones pendientes Mi cartera de acciones

	IDOperación	Tipo	Empresa	Cantidad	Precio
✘	994	COMPRA	BANCO POPULAR...	100	11.18

Si ya no deseamos que la operación se lleve a cabo antes de que haya sido procesada podremos pinchar en el aspa roja y cancelarla nos aparecerá el siguiente mensaje.



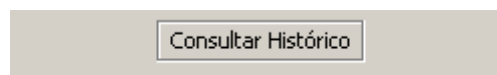
Una vez que se procesen las operaciones si pinchamos sobre Mi cartera de Acciones, y nos mostrará todas las acciones que tenemos en nuestra cartera.

Operaciones pendientes <input type="radio"/> Mi cartera de acciones <input checked="" type="radio"/>	
Empresa	Numero Acciones
BANCO POPULAR	100
ACCIONA	100
ABERTIS	100

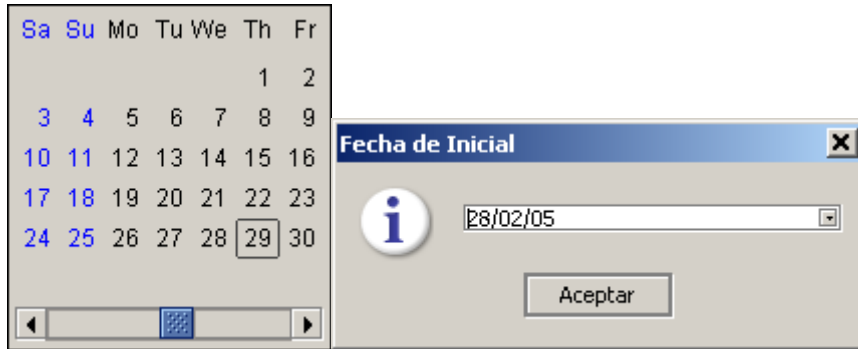
Por último nos queda la pestaña de “Información” en la cual consultaremos información sobre las empresas para poder conocer su actuación y sus esperanzas para el futuro.

Empresas	Información																	
<table border="1"> <thead> <tr> <th>Empresa</th> </tr> </thead> <tbody> <tr><td>TELEFONICA MOVILES</td></tr> <tr><td>BSCH</td></tr> <tr><td>ABERTIS</td></tr> <tr><td>BANKINTER</td></tr> <tr><td>FCC</td></tr> <tr><td>ENDESA</td></tr> <tr><td>TELEFONICA</td></tr> <tr><td>TPI</td></tr> <tr><td>METROVACESA</td></tr> <tr><td>FERROVIAL</td></tr> <tr><td>ALTADIS</td></tr> <tr><td>CINTRA</td></tr> <tr><td>TELECINCO</td></tr> <tr><td>BBVA</td></tr> <tr><td>ENAGAS</td></tr> <tr><td>NH HOTELES</td></tr> </tbody> </table>	Empresa	TELEFONICA MOVILES	BSCH	ABERTIS	BANKINTER	FCC	ENDESA	TELEFONICA	TPI	METROVACESA	FERROVIAL	ALTADIS	CINTRA	TELECINCO	BBVA	ENAGAS	NH HOTELES	<p>TELEFONICA MOVILES Sector: Tecnología/Telecomunicaciones Capital social: 2.125275448E9 Valor nominal de cada accion: 1.0 Número de ampliaciones de capital: 0</p> <p>Constituida en febrero del año 2000, en noviembre del mismo año realizó una oferta pública de suscripción que atrajo una fuerte demanda de inversores minoristas e institucionales. Las acciones de Telefónica Móviles cotizan en las bolsas españolas y en la New York Stock Exchange. Telefónica Móviles, líder en los mercados de España y Latinoamérica, es una de las diez primeras compañías mundiales de telefonía móvil. Se encuentra a la cabeza de la innovación tecnológica en el sector de telefonía celular y es la compañía líder en comunicación de datos y en desarrollos WAP, con productos y servicios de acceso (e-moción), contenidos a través del portal Terra Mobile junto a Terra Lycos y aplicaciones como la plataforma de pagos MobiPay.</p>
Empresa																		
TELEFONICA MOVILES																		
BSCH																		
ABERTIS																		
BANKINTER																		
FCC																		
ENDESA																		
TELEFONICA																		
TPI																		
METROVACESA																		
FERROVIAL																		
ALTADIS																		
CINTRA																		
TELECINCO																		
BBVA																		
ENAGAS																		
NH HOTELES																		

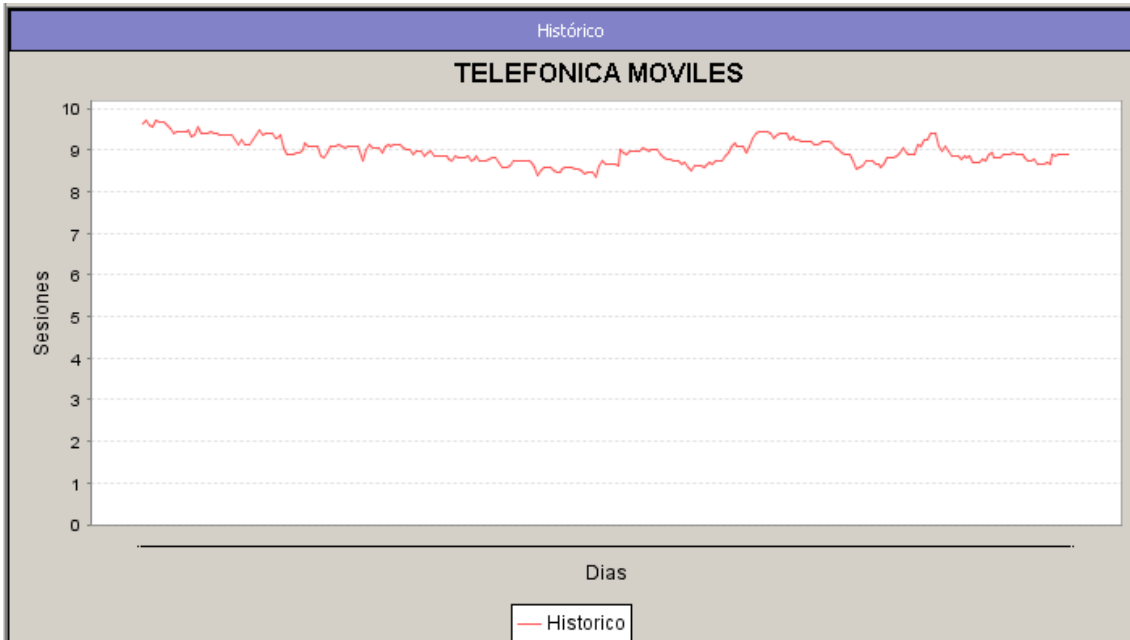
Obtendremos una descripción si simplemente la vamos seleccionando pero para ver sus movimientos históricos en la bolsa debemos pinchar en.



Nos preguntará el intervalo de fechas en los que queremos consultar el histórico mediante un calendario emergente

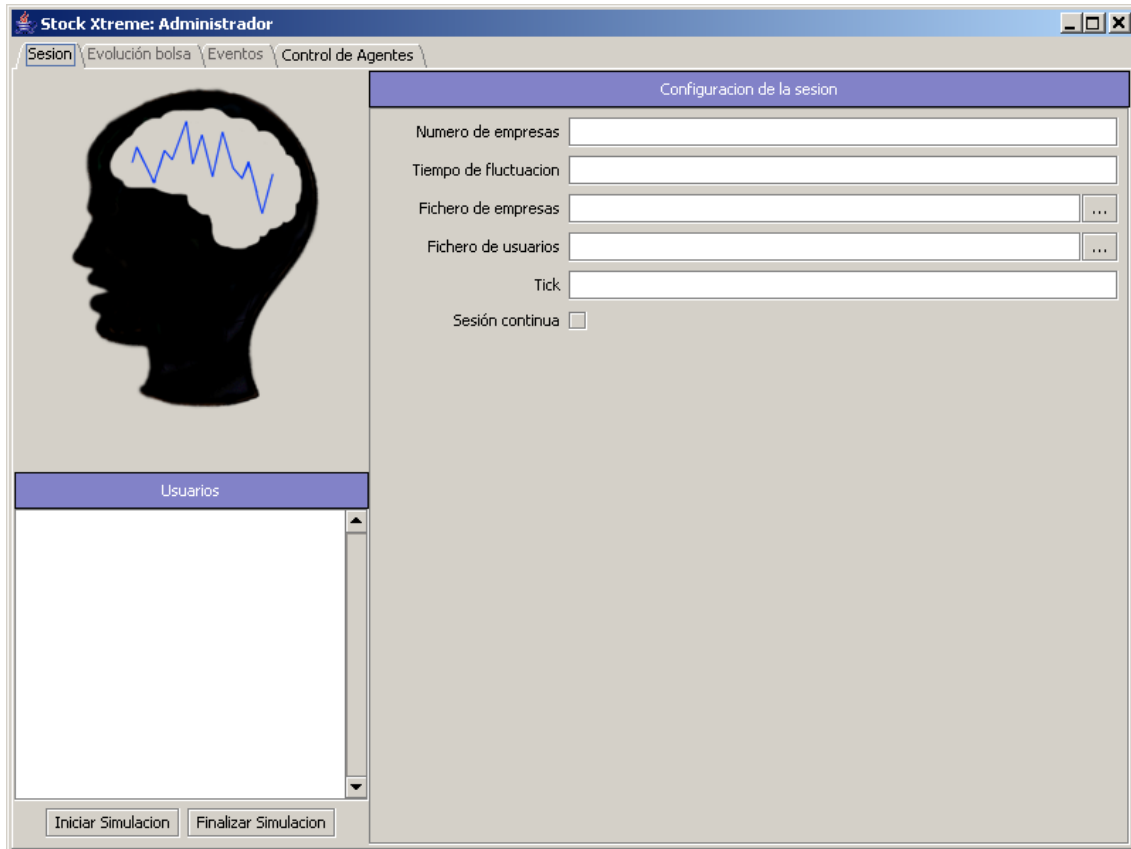


Una vez procesado los gráficos obtendremos el gráfico histórico del valor seleccionado.

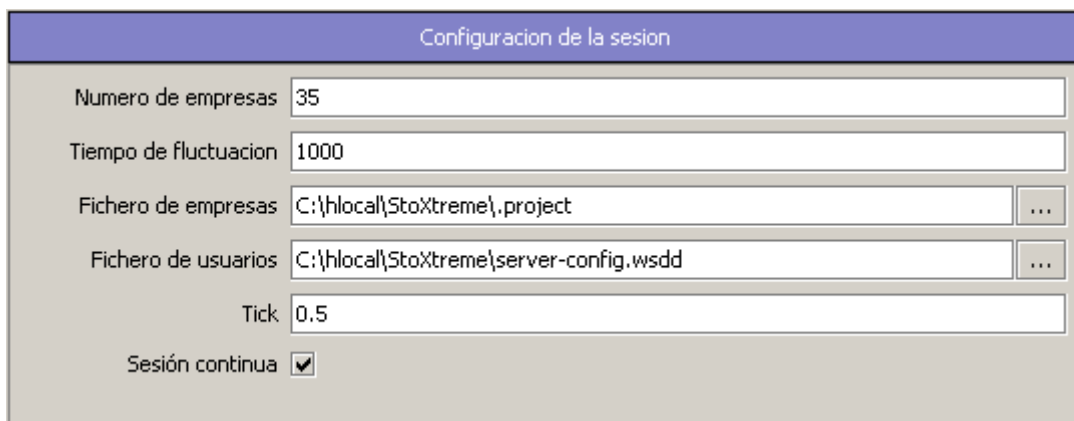


APENDICE C: MANUAL DE ADMINISTRADOR

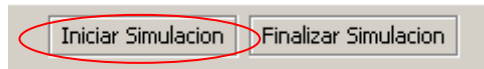
Este manual corresponde al encargado de conectar el sistema inicial de simulación bursátil y poder controlar a los usuarios y a los agentes que interactúan en él, inicialmente obtenemos la siguiente pantalla.



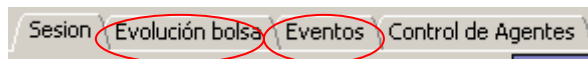
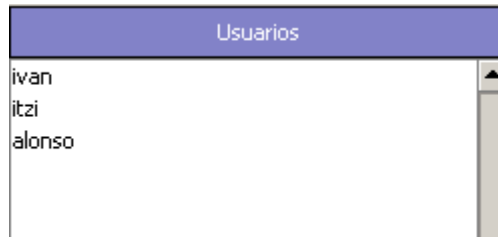
Debemos iniciar la simulación para ello podemos establecer los parámetros por defecto dejándolos en blanco o manualmente como en la siguiente imagen



Una vez que hemos introducido los parámetros adecuados procedemos a iniciar el servidor.



Una vez iniciado el servidor se generará los objetos bolsa y aparecerán los usuarios registrados que hay actualmente en el sistema y se habilitaran las pestañas de Evolución Bolsa y eventos.



En la pestaña de Evolución de la Bolsa nos muestra el precio de todas las acciones que existen en el mercado.

Precios en bolsa	
Empresa	Precio Actual
TELEFONICA MOVILES	9.95
BSCH	10.2
ABERTIS	16.72
BANKINTER	48.28
FCC	58.45
ENDESA	25.05
TELEFONICA	11.76
TPI	8.17
METROVACESA	67.1
FERROVIAL	55.6
ALTADIS	36.5
CINTRA	9.86
TELECINCO	16.56
BBVA	15.52
ENAGAS	16.1
NH HOTELES	13.37
UNION FENOSA	30.3
IBERDROLA	25.98
BANCO SABADELL	25.93
SOGECABLE	22.02
BANCO POPULAR	11.18
ANTENA3	17.21
ACERINOX	12.76
ARCELOR	37.25
REPSOL	20.36
IBERIA	1.92
GAS NATURAL	22.81
INDITEX	30.53
GAMESA	16.01
R.E.E	24.23
INDRA	14.91

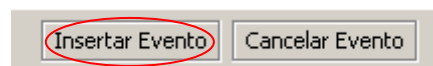
Y también se muestran los movimientos de todos los agentes y clientes que hay en la bolsa en tiempo real pudiendo organizar y consultar por cada campo de la siguiente tabla.

Log de Operaciones					
ID Operacion	ID Usuario	ID Empresa	Cantidad	Precio	Tipo Operacion
0	ivan#Agente1	IBERIA	651	1,94	Compra
1	ivan#Agente2	REPSOL	932	20,39	Compra
2	ivan#Agente3	ENDESA	292	25,06	Compra
3	ivan#Agent...	TELEFONICA	325	12,71	Compra
4	ivan#Agent...	IBERIA	642	2	Compra
5	ivan#Agent...	IBERIA	506	1,97	Compra
6	ivan#Agent...	INDITEX	68	30,81	Compra
7	ivan#Agent...	REPSOL	307	19,6	Compra
8	ivan#Agent...	FERROVIAL	712	55,96	Compra
9	ivan#Agent...	BANCO SAB...	621	25,69	Compra
10	ivan#Agent...	BANCO SAB...	469	26,47	Compra
11	ivan#Agent...	BANCO SAB...	297	25,94	Compra
12	ivan#Agent...	TELECINCO	78	17,17	Compra
13	ivan#Agent...	FERROVIAL	859	55,62	Compra
14	ivan#Agent...	TELECINCO	689	16,58	Compra
15	ivan#Agent...	R.E.E	449	25,01	Compra
16	ivan#Agent...	METROVACESA	700	67,12	Compra
17	ivan#Agent...	INDITEX	454	30,57	Compra
18	ivan#Agent...	NH HOTELES	819	13,42	Compra
19	ivan#Agent...	TELEFONICA...	382	9,77	Compra
20	ivan#Agent...	IBERIA	326	1,92	Compra
21	ivan#Agent...	ACERINOX	305	12,78	Compra
22	ivan#Agent...	ENAGAS	158	16,14	Compra
23	ivan#Agent...	FCC	819	56,62	Compra
24	ivan#Agent...	ENAGAS	347	15,26	Compra
25	ivan#Agent...	TPI	687	8,2	Compra
26	ivan#Agent...	R.E.E	286	24,26	Compra
27	ivan#Agent...	TELEFONICA	592	11,78	Compra
28	ivan#Agent...	BANCO SAB...	626	24,71	Compra
29	ivan#Agent...	INDRA	855	14,96	Compra
30	ivan#Agent...	INDITEX	440	30,57	Compra

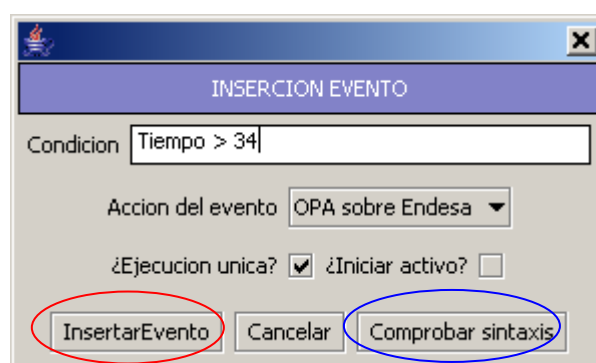
Ahora nos introducimos en la pestaña de Eventos para ver como el administrados puede ir informando al resto de usuarios, en tiempo real, de algún caso especial que se da en la bolsa para ello disponemos de unas variables

Variables del sistema	
Nombre	Valor
TIEMPO	27 ▲
PRECIO_TELEFONICA ...	10,8
PRECIO_BSCH	11
PRECIO_ABERTIS	18,17
PRECIO_BANKINTER	48,28
PRECIO_FCC	59,05
PRECIO_ENDESA	26,95
PRECIO_TELEFONICA	12,81
PRECIO_TPI	8,47
PRECIO_METROVACESA	74
PRECIO_FERROVIAL	58,15
PRECIO_ALTADIS	36,5
PRECIO_CINTRA	9,86
PRECIO_TELECINCO	18,21
PRECIO_BBVA	15,52
PRECIO_ENAGAS	16,5

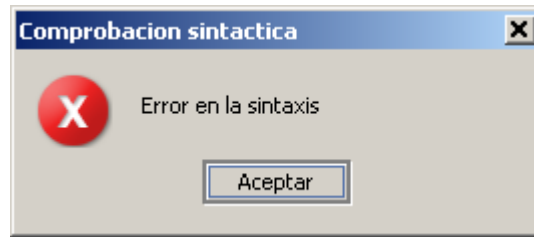
Cuando el administrador lo desee puede introducir un evento que puede ser de dos tipos uno que se ejecute una única vez u otro que se ejecute periódicamente



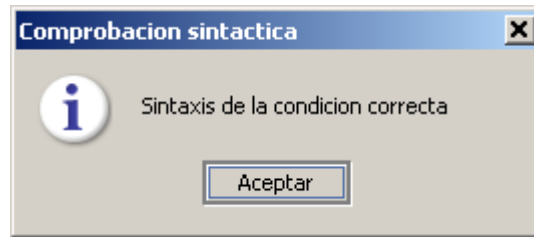
Nos aparece el siguiente recuadro y elegimos por ejemplo OPA sobre Endesa, establecemos una condición elegimos entre una vez o aviso periódico. Y damos a Insertar Evento podemos comprobar que la condición es correcta pinchamos en Comprobar sintaxis



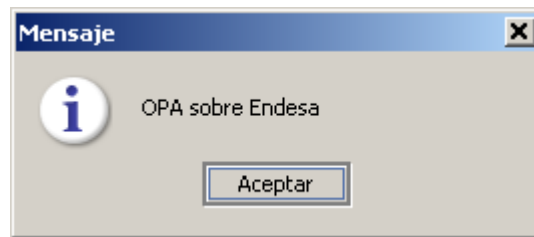
Si introducimos una Condición falsa nos indicara Stock Xtreme con el siguiente fallo.



En el caso que la condición sea correcta.

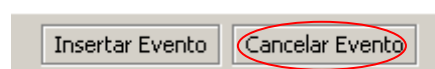


Una vez introducido el evento correctamente, cuando finalmente la condición se cumpla aparecerá el correspondiente mensaje al usuario.



En el caso que el administrado considere que el evento está obsoleto o ya no interesa podrá cancelarlo seleccionado el correspondiente evento y pinchando en Cancelar Evento

Control de Eventos		
Condicion	Accion	Activo?
TIEMPO > 10	OPA SOBRE ENDESA	<input checked="" type="checkbox"/>



Por último pinchamos en la pestaña de Control de Agentes para configurar los agentes su comportamiento y niveles psicológicos y sociales.

Tenemos por una parte los parámetros iniciales en el cual podremos elegir el numero máximo de Agentes el numero mínimo el número de Agentes el tiempo de espera la distribución para dicha espera etc...

The 'Parametros' window contains the following fields:

Parameter	Value
Numero máximo de agentes	100
Numero mínimo de agentes	6
Tiempo de espera mínimo	60
Tiempo de espera máximo	10
Distribucion del tiempo de espera	normal1
Gasto máximo	1000.0
Ratio respawn	0.1
Atenuacion rumor	0.1

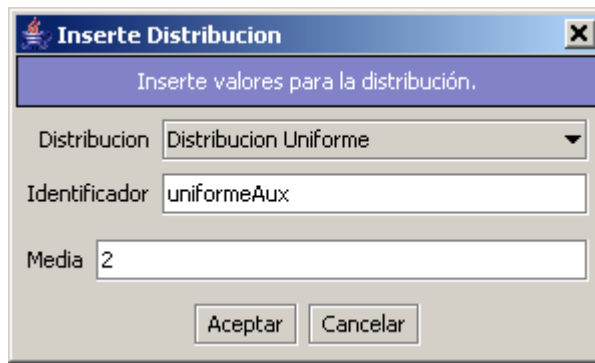
Ahora podremos configurar cada modelo agente con la siguiente ventana que está situada debajo de los parámetros.

The 'Modelo' window has tabs for 'Comportamientos', 'Social', 'Psicologico', and 'Distribuciones'. The 'Distribuciones' tab is active, showing a list of models:

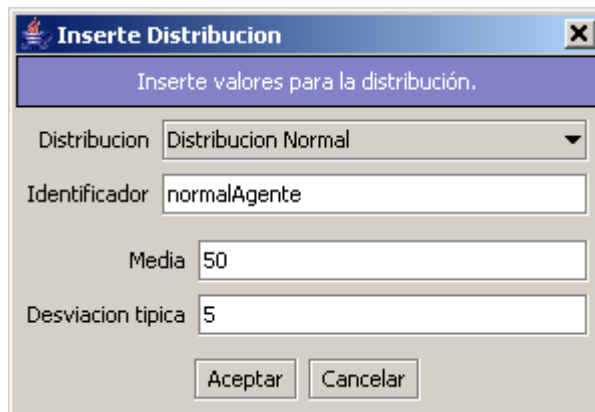
- normal1 - Normal(20.0,10.0)
- normal2 - Normal(50.0,20.0)
- normal3 - Normal(10.0,5.0)
- poisson1 - Poison(12.0,0.0)
- uniforme1 - Uniforme(10.0,30.0)

At the bottom, there are buttons for '+', '-', 'Cargar Fichero', and 'Guardar Configuración'.

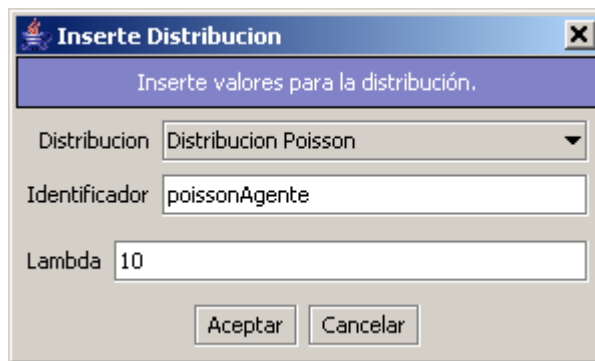
Veamos las distribuciones que siguen los agentes para sus tiempos de espera, si pinchamos en la pestaña distribuciones vemos que tenemos unas predeterminadas pero si queremos pinchando en el botón "+" nos aparecerá la siguiente ventana que tendremos que rellenar. Existen 3 tipos distintos:



The dialog box titled "Inserte Distribucion" contains the following fields: "Distribucion" is set to "Distribucion Uniforme", "Identificador" is "uniformeAux", and "Media" is "2". There are "Aceptar" and "Cancelar" buttons at the bottom.



The dialog box titled "Inserte Distribucion" contains the following fields: "Distribucion" is set to "Distribucion Normal", "Identificador" is "normalAgente", "Media" is "50", and "Desviacion tipica" is "5". There are "Aceptar" and "Cancelar" buttons at the bottom.



The dialog box titled "Inserte Distribucion" contains the following fields: "Distribucion" is set to "Distribucion Poisson", "Identificador" is "poissonAgente", and "Lambda" is "10". There are "Aceptar" and "Cancelar" buttons at the bottom.

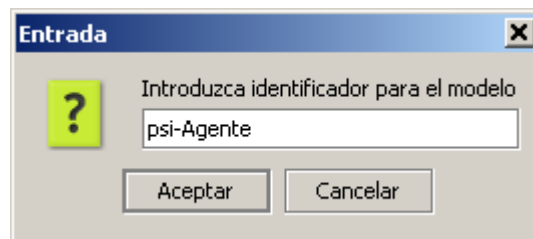
Si queremos eliminar alguna de las distribuciones requeridas pulsamos la distribución y pinchamos sobre el botón “-”.

Ahora pinchamos sobre el modelo psicológico de la misma manera que la anterior solo que para modificar los parámetros debemos realizar doble clic sobre el modelo insertado o el existente entonces nos aparecerá a la derecha la siguiente tabla.

Edición		
Parametro	Valor	Distribucion
tiempo_espera	normal1 - Normal(20.0,10.0)	<input checked="" type="checkbox"/>
numero_maximo_acciones_compra	poisson1 - Poison(12.0,0.0)	<input checked="" type="checkbox"/>
numero_minimo_acciones_compra	normal1 - Normal(20.0,10.0)	<input checked="" type="checkbox"/>
numero_maximo_acciones_venta		<input type="checkbox"/>
numero_minimo_acciones_venta	normal1 - Normal(20.0,10.0)	<input checked="" type="checkbox"/>
numero_maximo_cancelaciones		<input type="checkbox"/>
porcentaje_maximo_compra		<input type="checkbox"/>
porcentaje_maximo_venta	normal1 - Normal(20.0,10.0)	<input checked="" type="checkbox"/>
porcentaje_minimo_compra	poisson1 - Poison(12.0,0.0)	<input checked="" type="checkbox"/>
porcentaje_minimo_venta		<input type="checkbox"/>
porcentaje_subidaPrecio	normal3 - Normal(10.0,5.0)	<input checked="" type="checkbox"/>
porcentaje_bajadaPrecio	uniforme1 - Uniforme(10.0,30.0)	<input checked="" type="checkbox"/>
precio_recomendacion		<input type="checkbox"/>
precio_compra_recomendacion	normal3 - Normal(10.0,5.0)	<input checked="" type="checkbox"/>

Cada campo es rellenado con una de las distribuciones que hemos configurado anteriormente.

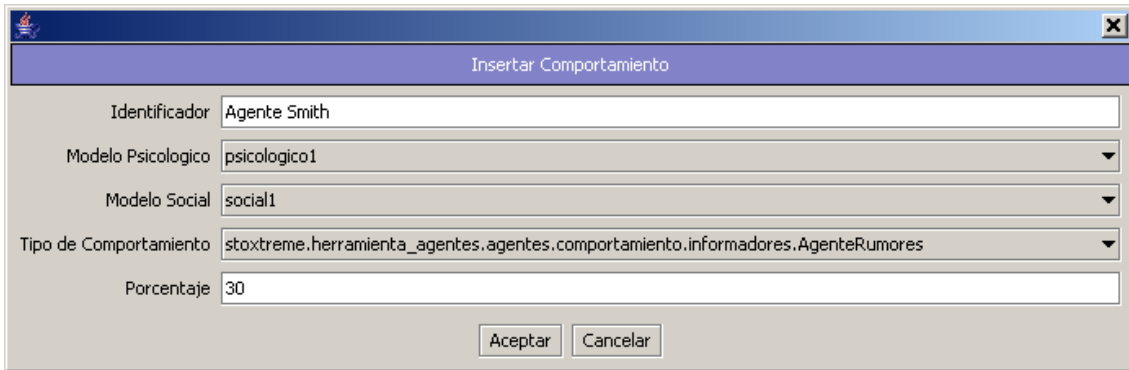
Para introducir un nuevo comportamiento psicológico pulsamos a “+” y posteriormente lo configuramos como antes. Para eliminar pulsa “-”.



El modelo social se maneja exactamente igual al modelo anterior salvo que los parámetros configurables son únicamente 2.

Edición		
Parametro	Valor	Distribucion
fiabilidad_rumor	poisson1 - Poison(12.0,0.0)	<input checked="" type="checkbox"/>
numero_conocidos	normal1 - Normal(20.0,10.0)	<input checked="" type="checkbox"/>

Dentro de modelo de “Comportamientos” vamos introduciendo diferentes comportamientos que engloban de los explicados anteriormente si pulsas “+” dentro de la carpeta raíz obtendremos la siguiente tabla a rellenar pondremos el identificador del Agente sus modelos y comportamiento y por ultimo el porcentaje Agentes de este tipo que habrá en sistema bursátil.

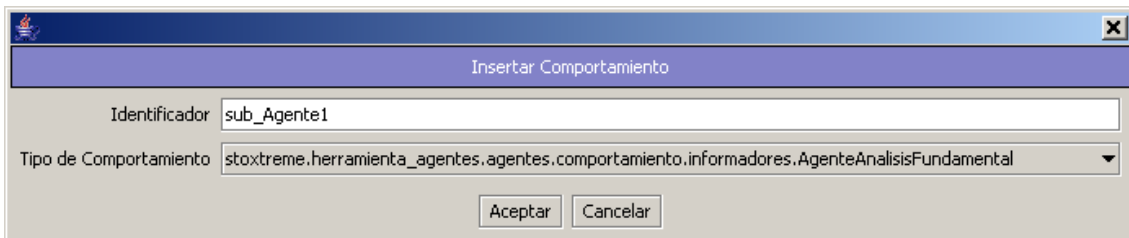


The screenshot shows a dialog box titled "Insertar Comportamiento". It contains the following fields and options:

- Identificador: Agente Smith
- Modelo Psicologico: psicologico1
- Modelo Social: social1
- Tipo de Comportamiento: stoxxtreme.herramienta_agentes.agentes.comportamiento.informadores.AgenteRumores
- Porcentaje: 30

At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Dentro de cada agente podrán crearse subagentes que tendrán los mismos modelos pero distinto comportamiento.

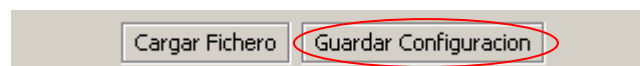


The screenshot shows a dialog box titled "Insertar Comportamiento". It contains the following fields and options:

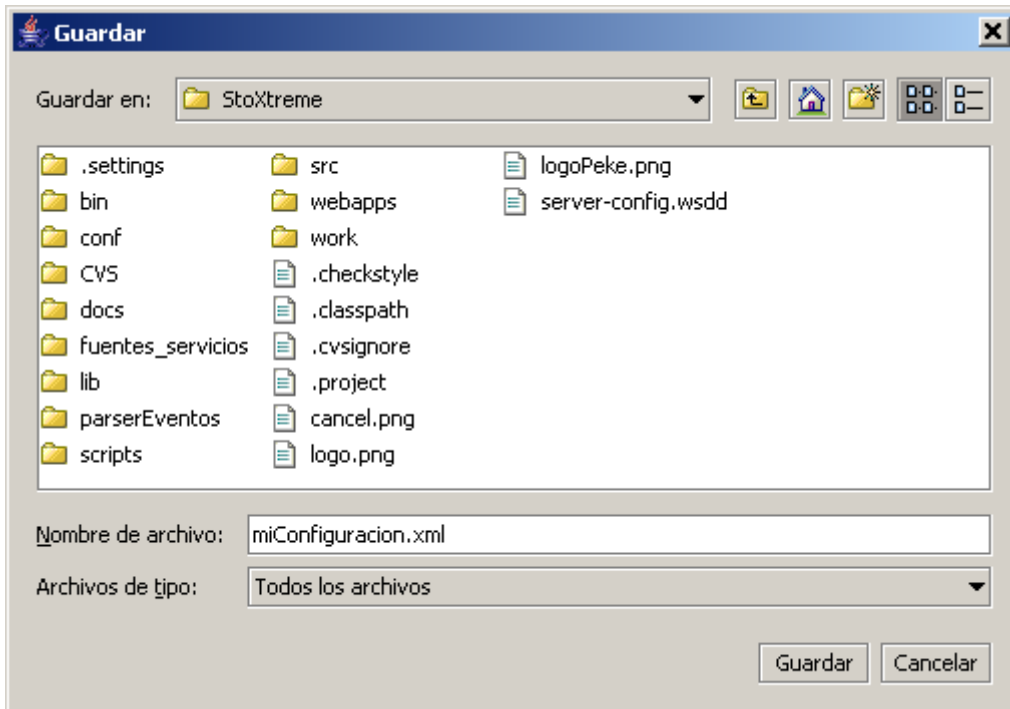
- Identificador: sub_Agente1
- Tipo de Comportamiento: stoxxtreme.herramienta_agentes.agentes.comportamiento.informadores.AgenteAnalisisFundamental

At the bottom, there are two buttons: "Aceptar" and "Cancelar".

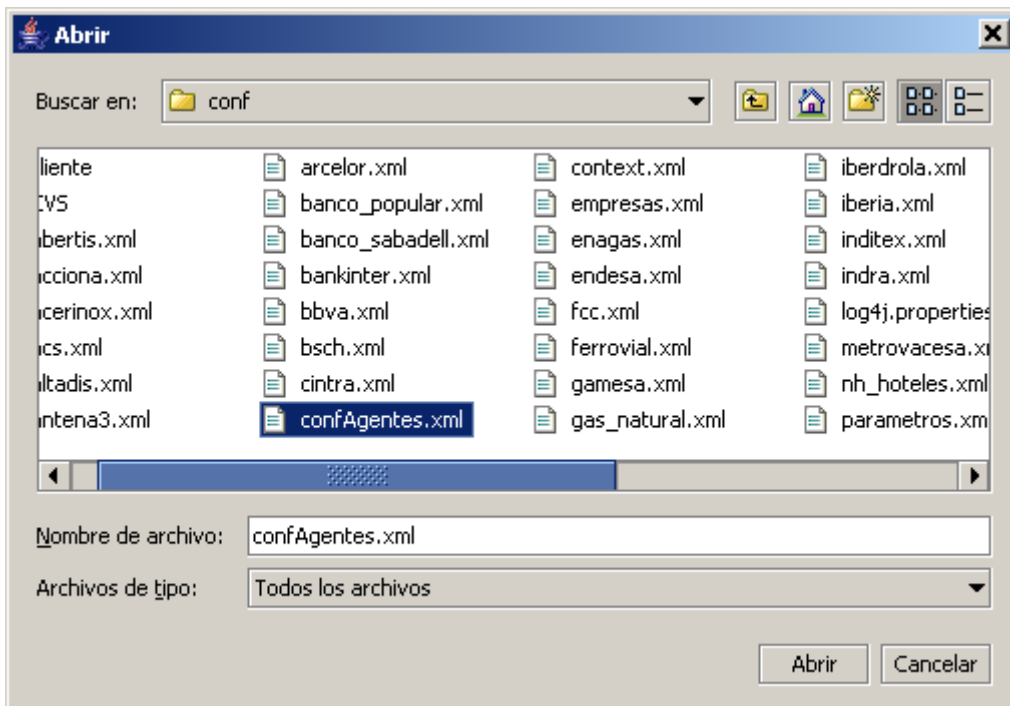
Por último cuando hayamos terminado de configurar los agentes Stock Xtreme nos permite guardar dicha configuración para posteriores usos ya que carga la configuración por defecto para guardar la configuración pulsamos.



Nos Aparecerá un navegador en que podremos seleccionar el directorio donde almacenar la configuración.



Y para cargar una configuración que ya ha sido almacenada pulsamos Cargar fichero y nos aparece el mismo navegador que nos permite cargar la configuración deseada.



APENDICE D: BIBLIOGRAFÍA

Libros

- [1] **Oriol Amat I Salas**. *La bolsa*. Deusto S.A. Ediciones, 2004.
- [2] **Vicenta Tena Rodríguez**. *Análisis de los mercados de valores : Una aproximación a su estructura y funcionamiento*. Dykinson, 2000.
- [3] **Bruce Eckel**. *Piensa en Java*. Prentice Hall 2005.
- [4] **Joshua Bloch**. *Effective Java : programming language guide*. Addison-Wesley, 2002.
- [5] **Ethan Cerami**. *Web services essentials*. Addison-Wesley, 2002.
- [6] **Theodore W. Leung**. *Professional XML development with Apache tools : Xerces, Xalan, Fop, Cocoon, Axis, Xindice*. Wiley, 2004.
- [7] **Ana Mas**. *Agentes software y sistemas multiagente : conceptos, arquitecturas y aplicaciones*. Prentice Hall, 2005.
- [3] **Joseph P. Bigus, Jennifer Bigus**. *Constructing intelligent agents using JAVA*. Wiley, 2001.

Páginas web

- **Bolsa de madrid (IBEX 35):**
<http://www.bolsamadrid.es/esp/contenido.asp?menu=4&enlace=/esp/indices/ibex35/ibex35.htm>
- **Yahoo Finances (IBEX 35):** <http://es.finance.yahoo.com/q?s=^IBEX>
- **Axis:** <http://ws.apache.org/axis/>
- **Tomcat:** <http://tomcat.apache.org/>
- **JavaCC:** <https://javacc.dev.java.net/>
- **JfreeChart:** <http://www.jfree.org/jfreechart/>
- **ServingXML:** <http://servingxml.sourceforge.net/>
- **MDateSelector:** <http://web.ukonline.co.uk/mseries/MDateSelector.html>

Artículos

- A builder's guide to Agent Based Financial Markets, Blake LeBaron, Brandeis University, February 2001
- Agent-based computational economics, Leigh Tesfatsion, Departament of Economics, Iowa State University, July 2002

- Building the Santa Fe Artificial Stock Market, Blake LeBaron, Brandeis University, June 2002
- The development of Ontology driven Multi-Agent Systems: A case study in the financial services domain, Anjalee Sujanani, Pradeep Ray, N. Paramesh, Ramprashad Bhar, University of New South Wales, Australia.

AUTORIZACION

Los abajo firmantes autorizamos a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria como el código, la documentación y/o el prototipo desarrollado.

Iván Gómez Edo

DNI: 02656962W

Itziar Pérez García

DNI: 47280080S

Alonso Javier Torres Ortiz

DNI: 02668860D